MERLIN for High Luminosity Large Hadron Collider Collimation

University of HUDDERSFIELD

Haroon Rafique

Centre for Doctoral Training in Next Generation Accelerators

International Institute for Accelerator Applications

University of Huddersfield

A thesis submitted to the University of Huddersfield
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

April 2017

Copyright

Abstract

The large hadron collider (LHC) uses the most energetic and highest luminosity man made proton beams on Earth. The high luminosity (HL-LHC) [1] upgrade aims to increase the levelled luminosity of the LHC by a factor of five, to $5 \cdot 10^{34} cm^{-2} s^{-1}$, by increasing the bunch population from 1 to $2.22 \cdot 10^{11}$ protons, and decreasing emittance, and β^*. Thus the stored beam energy increases from $\approx$ 362 MJ to $\approx$ 675 MJ per beam.

All synchrotrons encounter unavoidable proton losses. Protons that populate the beam halo pose a threat to the performance and lifetime of certain hardware, such as superconducting magnets, which in the LHC may be quenched by an impact of $\approx 1 \cdot 10^6$ protons [2]. A multi stage collimation system must operate at an efficiency such that no more than $2 \cdot 10^{-5}\%$ of protons incident on collimators may escape and impact upon these magnets [3].

To predict and protect against proton losses in the HL-LHC, collimation simulations must be performed. MERLIN, a C++ accelerator physics library, has been updated to carry out such simulations for the HL-LHC. Novel materials such as molybdenum graphite have been investigated as collimator materials, and a novel technique - collimation enhancement via a hollow electron lens (HEL) - has been studied. Using detailed simulations the performance and operation of possible collimation upgrades are explored.

Acknowledgements

Many opportunities were provided for me to present my work and learn from others, for support in this, his personal library, and continued guidance, I am very much obliged to my supervisor Roger Barlow.

I am grateful for the funding provided by the EPSRC through the Centre for Doctoral Training in Next Generation Accelerators (NGACDT). The NGACDT has offered many opportunities for growth, and I am grateful to the CDT cohort for our many escapades.

The MERLIN group constantly work towards an ever-growing accelerator physics library. I appreciate the hard work of all previous and current developers, in particular I am thankful for the guidance and collaboration of Roger Barlow, Rob Appleby, James Molson, Maurizio Serluca, and Sam Tygier.

This work would not be possible without fruitful collaboration with CERN. Specifically; Stefano Redaelli and Roderik Bruce for their guidance in collimation and the hollow electron lens, Adriana Rossi, Daniele Mirarchi, and Joschka Wagner for their partnership on the HEL and HL-LHC, Elena Quaranta for her assistance with novel and composite materials, Andrea Santamaria Garcia for her help with crab cavity failures, and Alessandra Valloni and Alessio Mereghetti for their work on composite materials and 6.5 TeV loss maps. Special thanks to Alessandra for calling me at all hours to provide constructive criticism. Many others must be thanked for making CERN an enjoyable place of work, and also the CERN cricket club for teaching me how to play.

I have picked the brains of many experts to perform this work, as well as those mentioned I thank Giulio Stancari for his expertise on HELs and Vladimir Shiltsev for his book on electron lenses.

Without the encouragement, distraction, and support received from my friends this endeavour would have been less enjoyable. My thanks to Anna Kolano, Caroline Montgomery, David Bruton, Jordan Taylor, and Roxana Rata for being a positive force in my life. Special thanks to Jordan and David, the poor souls who volunteered to proof read this thesis. I hope that you have learnt never to volunteer for anything ever again, and also not to repeat my mistakes.

I would also like to thank the members of the IIAA for their advice and support, especially Sue Kilcoyne and Becky Seviour.

Finally I am forever grateful to my family for their constant support, guidance, encouragement, and love.

To my parents and sister.

Contents

Total word count: 43,319

List of Figures

List of Tables

LIST OF TABLES

1

Introduction

1.1 Thesis Organisation

After introducing the LHC and HL-LHC in Chapter 1, the required accelerator physics and beam collimation is covered in Chapters 2 & 3. In Chapter 4 a detailed summary of the work done on the MERLIN C++ accelerator physics library is given, including some relevant physics. The focus of this thesis is developing aspects of MERLIN relevant to the challenges presented by the HL-LHC collimation system, namely; hollow electron lens collimation, treatment of composite and novel materials, long term tracking, and loss map production, such that MERLIN is a versatile tool. MERLIN is then benchmarked in Chapter 5, followed by the results of novel simulations in Chapter 6, including an integration study of the HEL in the HL-LHC. A summary and conclusion follows in Chapter 7.

1.2 LHC

1.2.1 Overview

The large hadron collider (LHC) was designed in the 1980's, approved in 1995, and built between 1998 - 2008. The purpose of the synchrotron is to address current mysteries in physics, including but not limited to:

- Explaining the origin of mass in the standard model of particle physics.
- Unifying of the fundamental forces.
- Investigating the nature of dark matter.
- Investigating matter antimatter asymmetry.

- Investigating early stages of the universe.

To achieve these goals, two counter rotating proton (or lead ion) beams are accelerated up to an energy of 7 $\frac{\text{TeV}}{\text{u}}$, and collide at one or more of four large multi-purpose detectors.

The LHC is split into octants, as shown in Fig. 1.1, it has a total length of 26.659 km but is not perfectly circular. It is split into 8 long straight sections (LSS), and 8 arcs, with dispersion suppressor (DS) regions between each arc and LSS.

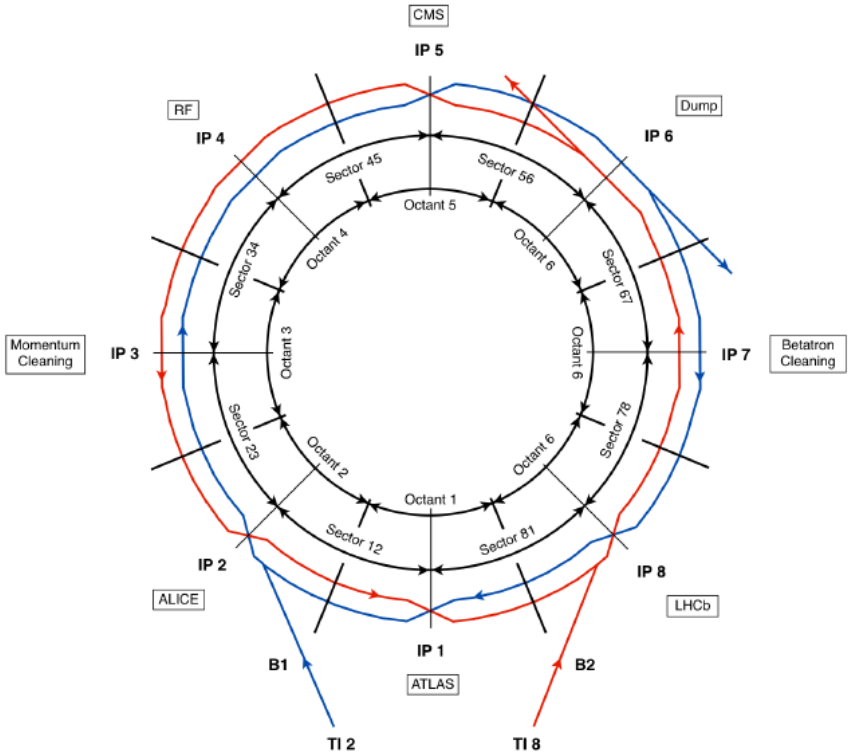

Figure 1.1: The LHC is split into octants, the interaction points are referred to as IPs and insertion regions as IRs, for example ATLAS is located at IP1, and the momentum collimation insertion is in IR3. From [4].

Two general purpose detectors exist; a toroidal LHC apparatus (ATLAS) and the compact muon solenoid (CMS) at IPs 1 and 5 respectively. The LHC beauty (LHCb) detector at IP 8 is an asymmetric detector that investigates matter antimatter asymmetry as well as other particle physics. A large ion collision experiment (ALICE) at IP 2 is primarily used for the lead-lead (Pb-Pb) collisions that take place for one month of every year of LHC operation. The four main detectors are shown in Fig. 1.2 along with

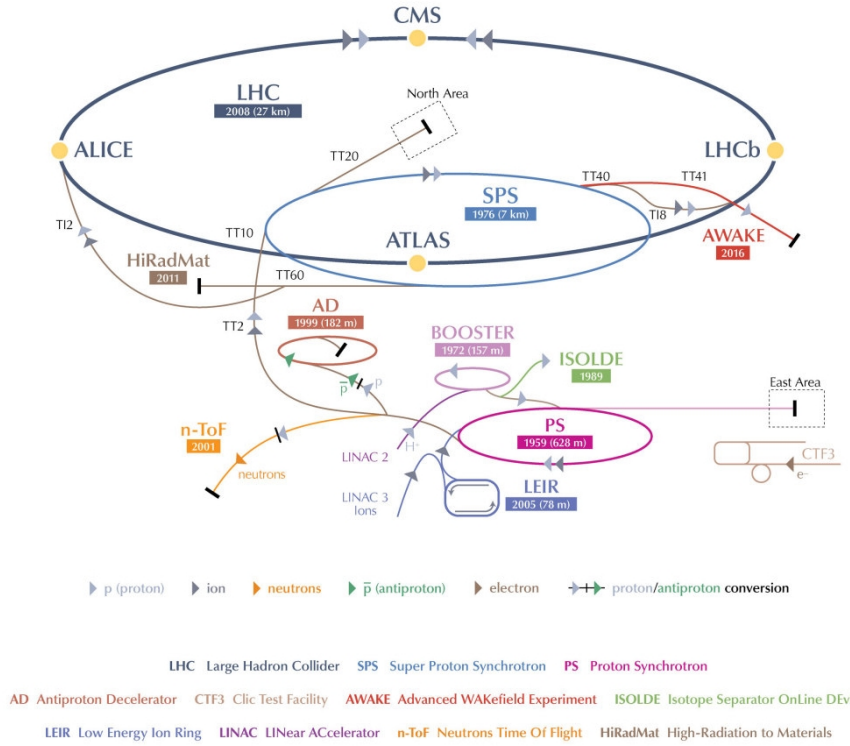

Figure 1.2: The CERN accelerator complex.

the accelerator complex. As well as these there exist smaller experimental insertions, including; total cross section, elastic scattering and diffraction dissociation (TOTEM), LHC forward (LHCf), and monopole and exotics detector at the LHC (MoEDAL).

The nominal operational parameters of the LHC are shown in Table 1.1. The LHC achieved first beams on 10^{th} September 2008, however there was a catastrophic failure on the 19^{th} September 2008, in which the superconducting (SC) interconnect between two dipole magnets that had not been correctly soldered, caused a quench. Superconducting magnets are operated at very low temperatures, a quench is the return of a SC magnet to a normal conducting state. As ≈ 10 GJ of energy is stored in the SC magnets, a fail-safe exists to remove the power to these magnets in case of a quench. This fail-safe acted as expected, however an electrical arc caused a breach in the high pressure supercooled helium vessel, which resulted in a pressure explosion with enough force to break the 10 tonne magnets from their mountings, and caused contamination and damage in a large section of the machine.

As a result of the disaster, Run I of the LHC was set back to November 2009.

On 30^{th} November 2009, the LHC reached a milestone in circulating 1.18 TeV beams, beating the previously held record of 0.98 TeV at the Tevatron [5]. An energy of 3.5 TeV per beam was reached in early 2010, in November 2010 the first proton run ended and the first lead ion run began. After running at 4 TeV from 2011, long shutdown 1 (LS1) was postponed until early 2013 due to the announcement by ATLAS and CMS, of the discovery of a 'Higgs like' particle at 126 $\frac{\text{GeV}}{\text{c}^2}$.

LS1 began in February 2013, and involved general maintenance and upgrades of the LHC and its injectors. A consolidation of all magnet interconnects was also performed in order to prevent failures similar to the one in 2008. To operate at higher beam energies, the SC magnets must be 'trained' to reach higher currents and thus magnetic fields. This is a slow process and lead to the decision to start Run II at 6.5 TeV rather than the nominal 7 TeV. On the 3^{rd} of June 2015 Run II began at a collision energy of 13 TeV, the proton run ended in November when the lead ion run began, and a record energy of 1 PeV was reached for lead ions.

The projected peak and integrated luminosity of the LHC is shown in Fig. 1.3, at the end of 2015 the LHC had reached an integrated luminosity of ≈ 30 fb^{-1} [6], and a peak luminosity of $\approx 5{\cdot}10^{33}$ cm^{-2}s^{-1} [6] at ATLAS and CMS.

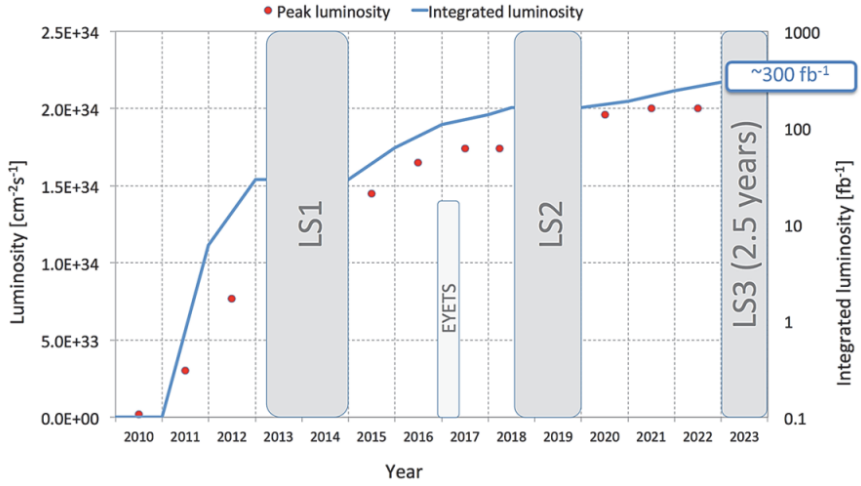

Figure 1.3: Possible peak and corresponding integrated luminosity up to the 'ultimate' limit, including long shutdowns (LS) 1 to 3, and the end of year technical stop (EYETS) in 2016. From [7].

1.2.2 Collimation

In order to obtain the maximum experimental data, the LHC must remain operational. To reduce downtime a complex machine protection system exists, at the heart of which is the collimation system; a hierarchy of beam targets that intercept the beam halo in order to reduce proton losses in superconducting magnets and other sensitive hardware.

In the transverse beam plane (xy in later defined co-ordinates), accelerator beams are generally Gaussian in shape. The beam profile of the LHC is observed to be Gaussian; ≈ 67.3 % of the particles are contained within one standard deviation, or σ of the bunch. As depicted in Fig. 1.4, the core of the beam is generally accepted to be the particles within 3 σ. As the LHC beam is Gaussian, the core contains ≈ 99.7 % of the particles, leaving ≈ 0.3 % in the beam halo.

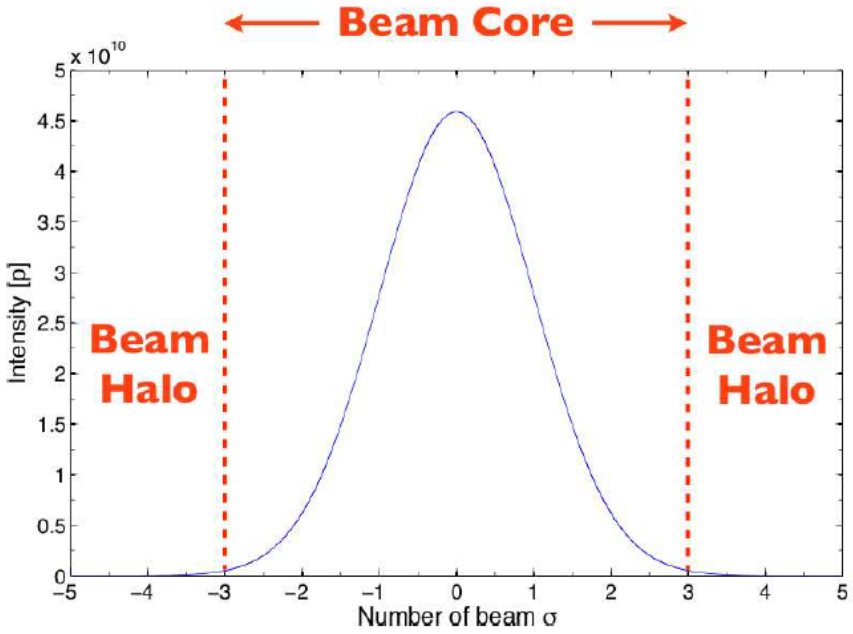

Figure 1.4: The LHC beam core and halo in transverse phase space. From [4].

Quench limits for LHC dipoles were previously thought to be of the order $\approx$ 10 mW cm^{-3}, however more recent studies have estimated up to 47 mW cm^{-3} [8]. This equates to a tiny fraction of the LHC stored beam energy of 362 MJ at nominal settings.

The collimation system must provide a cleaning efficiency of 99.99992 %, *i.e.* this proportion of proton losses in the LHC must be in collimators rather than other components. To achieve this a multi-stage collimation system operates, as well as injection and extraction protection, and protection of the inner triplet magnets (quadrupoles

that focus the beam at collision points). A full description of the collimation is given in chapter 3. One of the expected performance limitations of the LHC is the cleaning efficiency at nominal energy [9].

1.3 High Luminosity LHC

The high luminosity LHC upgrade (HL-LHC) aims to improve the performance of the LHC by increasing the machine luminosity, defined in equation 1.1, by a factor of 5, and thus the integrated luminosity by a factor 10 [9].

$$\mathcal{L} = \gamma \frac{n_b N^2 f_{rev}}{4\pi\beta^* \epsilon_n} R, \tag{1.1}$$

the geometrical reduction factor R is given by equation 1.2;

$$R = \frac{1}{\sqrt{1 + \frac{\theta_c \sigma_z}{2\sigma}}}, \tag{1.2}$$

where:

γ is the Lorentz gamma factor,
n_b is the number of machine bunches,
N is the number of particles in a bunch,
f_{rev} is the revolution frequency of a bunch in the machine,
β^* is the beta function at the interaction point,
ϵ_n is the normalised transverse emittance of the beam,
θ_c is the full crossing angle between colliding beams,
σ is the transverse r.m.s. beam size, and
σ_z is the longitudinal r.m.s. beam size.

The increase in luminosity will be achieved by;

reducing the beam emittance ϵ,
increasing the bunch population N,
decreasing the bunch spacing to 25 ns,
reducing the interaction point β^* using large aperture NbTi separator and Nb_3Sn inner triplet magnets,
reducing the crossing angle θ_c with the use of crab cavities,
preparing the machine for collision debris and radiation.

Parameter [unit]	LHC	HL-LHC 25 ns	HL-LHC$_{BCMS}$ 25 ns	HL-LHC 50 ns
Collision energy [TeV]	7	7	7	7
Number of particles per bunch N_b [-]	$1.15{\cdot}10^{11}$	$2.2{\cdot}10^{11}$	$2.2{\cdot}10^{11}$	$3.5{\cdot}10^{11}$
Number of bunches per beam n_b [-]	2808	2748	2604	1404
Total number of particles N_{tot} [-]	$2.3{\cdot}10^{14}$	$6{\cdot}10^{14}$	$5.7{\cdot}10^{14}$	$4.9{\cdot}10^{14}$
Beam current [A]	0.58	1.09	1.03	0.89
Crossing angle [μrad]	285	590	590	590
Beam separation [σ]	9.4	12.5	12.5	11.4
Beta function at collision β^* [m]	0.55	0.15	0.15	0.15
Normalised emittance ϵ_n [μm]	3.75	2.5	2.5	3
Bunch length r.m.s [m]	$7.55{\cdot}10^{-2}$	$7.55{\cdot}10^{-2}$	$7.55{\cdot}10^{-2}$	$7.55{\cdot}10^{-2}$
Geometric luminosity factor (no crab cavities) R_0 [-]	0.836	0.305	0.305	0.331
Geometric luminosity factor (with crab cavities) R_{cc} [-]	(0.981)	0.829	0.829	0.838
Peak luminosity (without crab cavities) $[cm^{-2}s^{-1}]$	$1{\cdot}10^{34}$	$7.55{\cdot}10^{34}$	$7.55{\cdot}10^{34}$	$7.55{\cdot}10^{34}$
Virtual luminosity (with crab cavities) $L_{peak} \cdot \frac{R_0}{R_{cc}}$ $[cm^{-2}s^{-1}]$	$(1.18{\cdot}10^{34})$	$19.54{\cdot}10^{34}$	$18.52{\cdot}10^{34}$	$21.38{\cdot}10^{34}$
Levelled luminosity $[cm^{-2}s^{-1}]$	-	$5{\cdot}10^{34}$	$5{\cdot}10^{34}$	$2.5{\cdot}10^{34}$

Table 1.1: High luminosity and nominal LHC parameters. From [9].

1.3.1 Overview

In order for the HL-LHC to be attainable an upgrade of the entire injector chain is also under way; the LHC injector upgrade (LIU) project should be completed in LS2 (2019) [9] at a cost of ≈ 186 million Swiss francs [10], the HL-LHC project has a projected cost of ≈ 830 million Swiss francs. As well as this upgrades are planned for ATLAS and CMS between LS2-LS3, and for LHCb and ALICE in LS2 [9].

Table 1.1 details the possible HL-LHC parameters for various configurations, including 25 or 50 ns bunch spacing, and the BCMS regime with very small emittance.

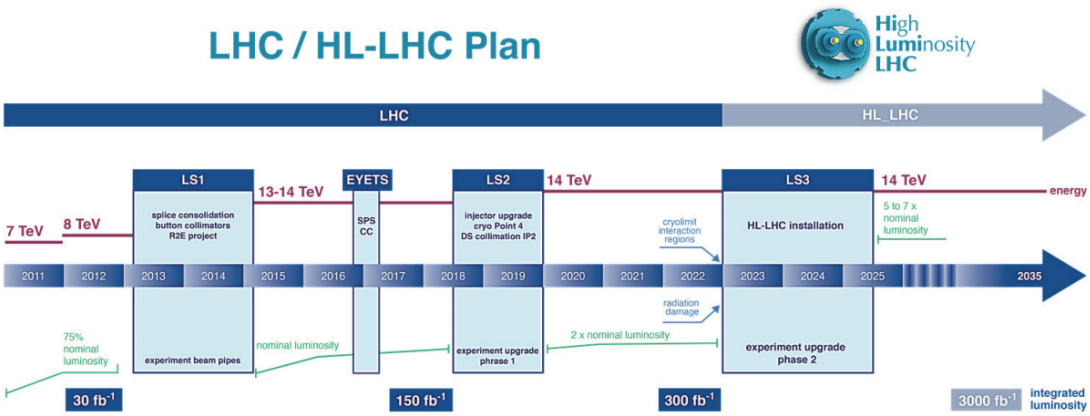

Figure 1.5: The LHC baseline plan including luminosity, energy, and the purposes of each long shutdown. From [9].

The systems of the LHC that are expected to reach the end of their lifetime by LS3, or require modification to enable luminosity beyond the LHC design, are summarised

below [9]:

Inner Triplet Magnets: Some triplet quadrupoles and corrector magnets will be radiation damaged after receiving a dose of ≈ 30 MGy after 300 fb^{-1}. In order to avoid sudden failures, replacement must take place before sufficient damage.
Cryogenics: A new cryogenics plant will be installed in IP4 to allow interventions in the inner triplet regions without warming up the arc section magnets.
Collimation: As well as upgrading the majority of the collimation system with novel materials to withstand the increase in luminosity and decrease impedance, additional collimators are required in the dispersion suppressor (DS) regions where off-momentum particles have been observed to be lost. This is particularly important for fragmented ions in IP2, and all particles after the collimation insertions in IP3 and IR7. A DS collimator must be cold, thus it is proposed to replace a standard LHC dipole (8.3 T field, 14.2 m length) with a high field magnet of shorter length (11 T field, 11 m long), thus making room for a special collimation device.
Cold Powering: To improve availability, some electronic devices (power supplies and associated equipment) will be moved out of the tunnel to protect them from radiation. Current electronic boards will be replaced with radiation hardened ones to improve robustness. New materials for superconducting links, such as YBCO, Bi-2223, and MgB_2 will allow power converters to be moved further from the machine.
Quench Protection System: The design of this system is almost two decades old and thus it will be updated.
Machine Protection: The injection kicker system, injection protection collimators (TDIs), and interlock systems require renovation post 2020.
Remote Manipulation: As elements of the machine will become radiation damaged, a system is required for replacing parts such as collimators safely. This is likely to involve robotics and remote manipulation.

The project is organised into work packages (WPs), as shown in Fig. 1.6.

1.3.2 WP2: Accelerator Physics

1.3.2.1 Reducing Beta*

There are three main limitations to decreasing β^*, assuming replacement of the inner triplet only [9]:

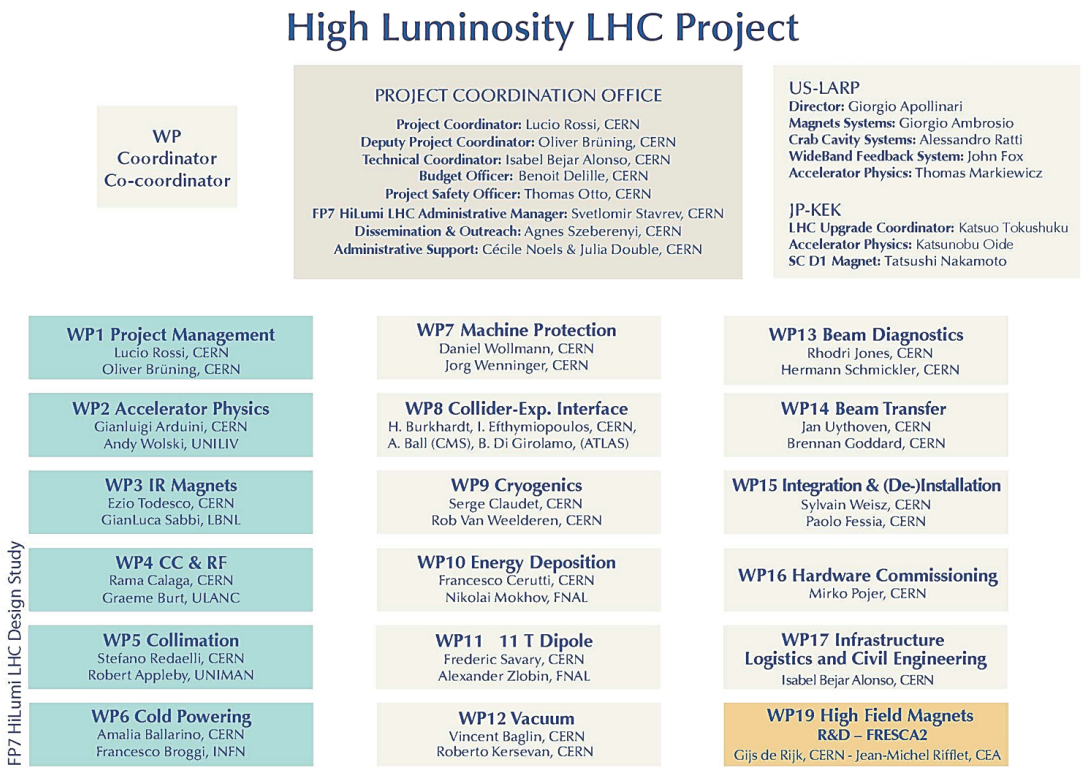

Figure 1.6: The organisation of HL-LHC work packages. From [9].

- mechanical acceptance of the existing matching section,
- gradient limits of the matching quadrupoles,
- strength limits of the arc sextupoles.

It was found that β^* is limited to 30 cm by the aperture and material of the SC triplet quadrupoles, this was estimated to be reduced to $\approx$ 25 cm when using higher gradient Nb_3Sn technology [9]. To reach the required β^* for the HL-LHC, the only solution is to use larger aperture (150 mm) triplet magnets, and D1 (first separation/recombination dipole).

A novel optics configuration of the LHC, the achromatic telescope squeezing (ATS) scheme has been developed in order to decrease the β^* for the HL-LHC. This involves using the arcs adjacent to the respective IP as matching sections. As a result the β-beating waves generated in these adjacent arcs reach their maximum at every other sextupole, thus increasing their chromatic correction efficiency, and removing the third limitation [11].

The ATS scheme is partly compatible with the nominal LHC, and was tested in

machine development (MD) studies in 2011 [11] and 2012 [12]. Operationally the ATS scheme has two stages [11]:

(i) **pre-squeeze**: a standard squeeze using the matching quadrupoles in the high luminosity insertions (IR1 and IR5).

(ii) **telescoping squeeze**: a further reduction in β^* (typically by a factor 4-8) by acting only on arcs adjacent to IP1 and IP5 (*i.e.* arcs 81, 12, 45, and 56), in which β-beating waves at the position of sextupoles results in an increase in effectiveness of chromaticity sextupoles in these arcs.

1.3.2.2 Approaches to Luminosity Levelling

In order to operate within pile-up limitations of the detectors, the HL-LHC will operate with a levelled luminosity much lower than the possible peak luminosity, as shown in Fig. 1.7. This will allow a sustained luminosity for each 'physics' part of the LHC operational cycle.

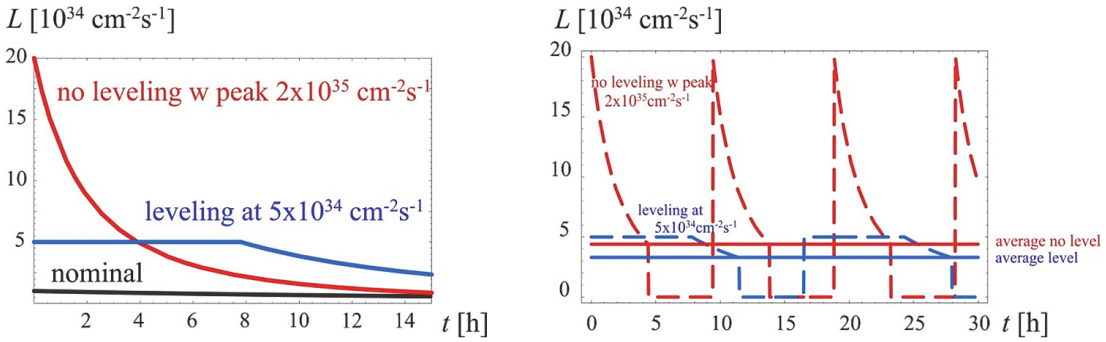

Figure 1.7: Levelled luminosity in the HL-LHC. The left hand plot compares luminosity lifetimes of the nominal LHC (black), virtual peak HL-LHC (red), and intended levelled HL-LHC (blue). The right hand plot compares the average luminosities for the virtual peak HL-LHC (red) and intended levelled HL-LHC (blue). From [9].

There are multiple approaches to achieving luminosity levelling, mostly utilising the relationships in equation 1.1, one or more variables may be modulated over time, once the required luminosity is attained, in order to compensate for the loss of particles N in collisions and other mechanisms:

1. Decreasing the crossing angle θ_c.
2. Decreasing the bunch length σ_z.
3. Decreasing β^* by focussing the beam.

4. Varying the distance between the two colliding beams.
5. Changing the crab cavity (CC) radio-frequency (RF) voltage.

The nominal LHC target for integrated luminosity is around 40 fb^{-1} per year, the HL-LHC has a target of $\approx$ 250 fb^{-1}.

1.3.3 WP4: Crab Cavities & RF

At the collision points of the LHC, bunches that are spaced by either 50 or 25 ns travel in the same vacuum chamber. In this part of the machine it is possible for 'long-range beam-beam' effects or 'parasitic' collisions to take place as well as the desired 'head-on' collisions, as shown in Fig. 1.8. Parasitic collisions are collisions between particles in bunches other than those in the current head-on collision. The beam-beam effect refers to the electromagnetic forces that occur between bunches that travel in close proximity. These forces are complex, non-linear, may be time dependent, and result in a major limit for all colliders.

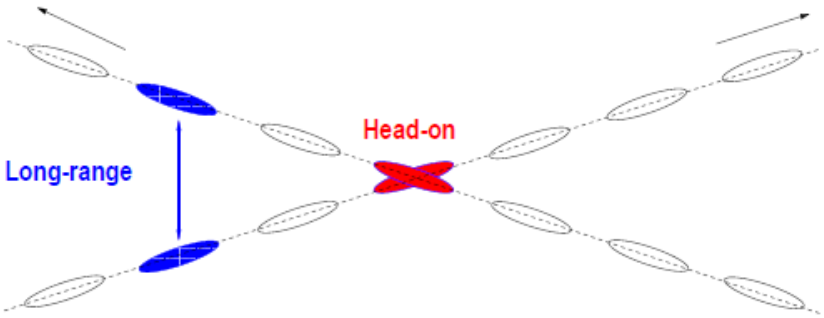

Figure 1.8: Long range beam-beam interaction and head-on collisions in a particle collider experiment. From [13].

The minimum separation between each beam is limited by the beam-beam interaction and parasitic collisions, for the HL-LHC it is larger (in σ) than in the LHC (Table. 1.1. This results in a larger crossing angle and thus reduction in luminosity at collision points. One major tool in the goal of achieving higher luminosity is the ability to reduce the geometric reduction factor (equation 1.2). Crab cavities (CCs) are novel RF deflecting cavities that are used to rotate particle bunches in the longitudinal plane in order to compensate for the geometric luminosity loss at the IPs.

Due to limited separation between beam pipes near the main detector IPs (194 mm), a global CC scheme was investigated, where a single CC or CC insertion was used per

beam in an area with larger pipe separation (near IP4 420 mm is available) which would allow for larger CCs. It is clear that the position of all CCs depend on the phase advance of the bunch, and it was found that a global scheme was incompatible with the two crossing schemes used in IP1 (vertical) and IP5 (horizontal) [9].

Instead the option of local CC insertions was deemed more practical, despite the minimal beam pipe separation of 192 mm. This way crabbing of the beam is performed immediately prior to the IP, and is corrected immediately post-IP. This option requires more CCs and for the CCs to be compact [9]. In total 2-4 CCs are required per IP and per beam, resulting in a total of 8-16 CCs in the HL-LHC.

1.3.3.1 Crab Cavity Failures

CCs have never been used in a hadron machine before, so experience must be taken from the KEKB collider [14] at KEK in Japan. CC failures are amongst the most devastating failure cases for the HL-LHC as simulations have shown that the beam may be kicked transversely by a number of σ in the ultra-fast regime (*i.e.* before a beam dump can be triggered) [9]. The LHC beam dump system can normally be triggered (by significant losses in beam loss monitors (BLMs) or other interlocks) after 3 turns. The beam loss system has a detection time of 40 μs, however the time between failure detection and a beam dump is of the order of 200 μs, as shown in Fig. 1.9. Each LHC turn is 80 μs, meaning that the time between detection of a failure and a beam dump trigger in the LHC is 3 turns. The BLM interlock is the quickest in response time, for comparison the collimation system interlocks have a detection time of $\approx$ 1 ms.

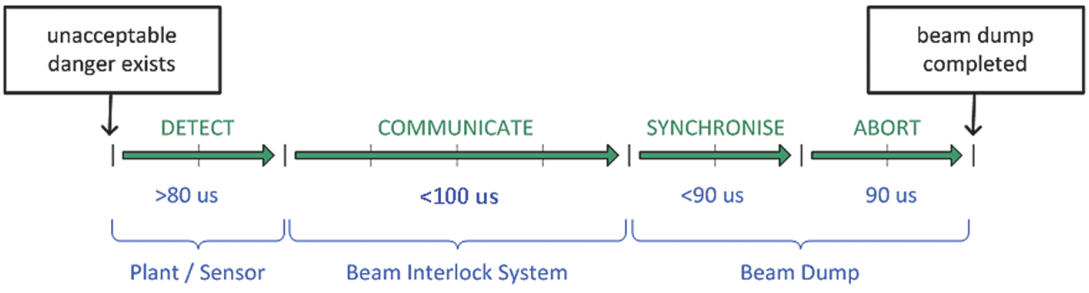

Figure 1.9: Machine protection system (MPS) response time from detection to beam dump. From [9].

CC failures can occur on different time scales, slow failures such as power or cryogenics issues are expected to take longer than 15 ms, which is not of major concern for the MPS. A CC quench, an arc in the CC coupler, or a CC failure due to multipacting

however may occur in less than a single turn. Quenches for example may cause a voltage decay or phase shift that can cause an overpopulation of the bunch tails before the BLM system can trigger a beam dump, corresponding to ≈ 30 MJ of stored energy [9]. The collimation system is designed to withstand only 1 MJ in case of accidental fast losses. CC failures are more devastating if more than one cavity fails at the same time. A number of mitigation options are being investigated, such as new tools for faster detection of such failures, and the installation of removable absorbers.

To protect against CC failures and for improved machine protection and collimation of the HL-LHC beam, active halo control, *i.e.* a method of controlling the halo population, is an area of great interest. If one can deplete the halo in a controllable way, such failures may cause less damage.

1.3.4 WP5: Collimation

As one of the performance limitations of the LHC is collimation and machine protection for beams with a stored energy of 362 MJ, it is clear that for the HL-LHC a significant collimation upgrade is required. This upgrade involves improved design concepts, novel collimator materials, dispersion suppressor collimation, and advanced collimation methods.

Collimators consist of movable jaws of a given material, that are inserted into the beam in the transverse xy axis to intercept halo particles. Possible improved designs for the HL-LHC include a rotatable collimator that has numerous faces, developed at SLAC [15] and shown in Fig. 1.10. If a face is damaged, instead of removing and replacing a conventional collimator or jaw, the novel collimator could simply be rotated, thus offering robustness and decreasing down time.

Another improvement developed for the HL-LHC is the addition of button shaped beam position monitors (BPMs) that are embedded into the face of the collimator jaw at the extremities in order to align the collimators quickly [16]. The embedded BPM button, shown in Fig. 1.10, is now the baseline for all collimators and a number of new collimators were installed in LS1 with the BPM technology.

1.3.4.1 Novel Materials

Due to the increased stored beam energy in the HL-LHC new materials are under investigation, with the following driving properties [9]:

- low resistive wall impedance to increase beam stability,
- high cleaning efficiency for effective collimation,

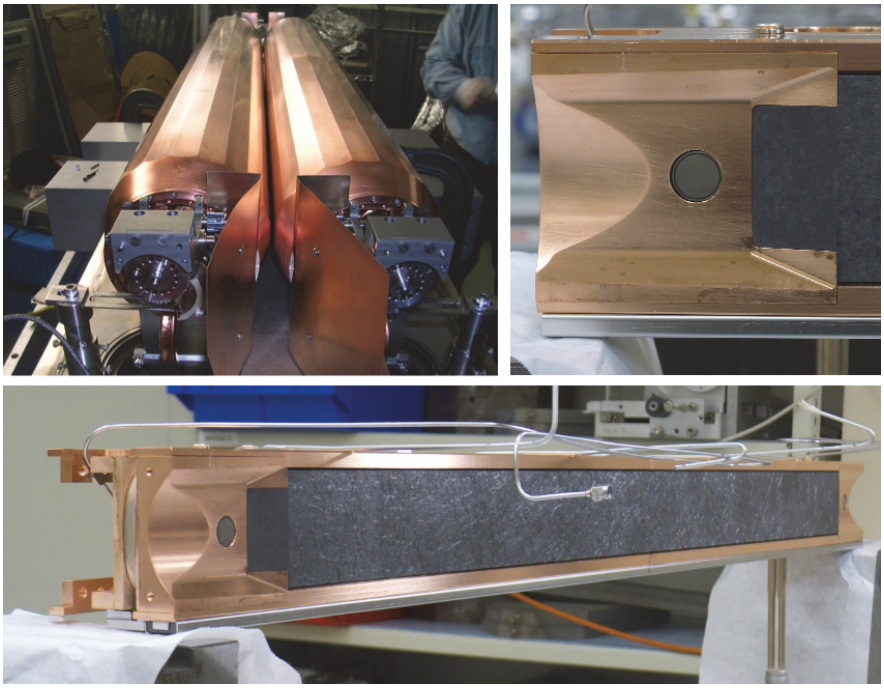

Figure 1.10: Rotatable collimator prototype (top left), and button BPM embedded in collimator jaws (top right, bottom). From [9].

- high geometrical stability to maintain precision under thermal stress,
- high structural robustness to cope with worst case scenarios (*e.g.* injection kicker failure).

Two metal-carbon composites have been identified as options as they combine the electrical and mechanical properties of metals with the thermal advantages of carbon allotropes (graphite and diamond). The two best candidates are copper-diamond (CuCD) and molybdenum-graphite (MoGr). As with other collimator materials, mechanical strength may be improved with the use of carbon fibres, and coatings such as pure molybdenum may be used to decrease the impedance.

1.3.4.2 Dispersion Suppressor Collimation

It has been observed in proton and ion operation of the LHC that significant losses occur in certain DS regions, particularly the DS regions adjacent to the collimation insertions (IR3 and IR7), and collision IPs (IP1,2,4, and 8). This poses problems as there is a lack of available space in the DS regions, and collimation may require a collimator compatible with cryogenic operation. Initially an LS1 intervention was

planned, to move 24 SC magnets per IR, this was a massive modification to the LHC, and was deemed unnecessary after evaluating collimation performance from 2011 and 2012. Instead the longer time scale allowed the development of 11 T dipole magnets so that a collimator may be integrated between the SC dipoles without affecting the machine optics. Replacing a 14.2 m long 8 T SC dipole with two high field 6.257 m long 11 T magnets allows 2.4 m for a warm collimator. This solution is more practical as any SC dipole in the LHC may be replaced with the double 11 T dipole plus collimator combination, as shown in Fig. 1.11.

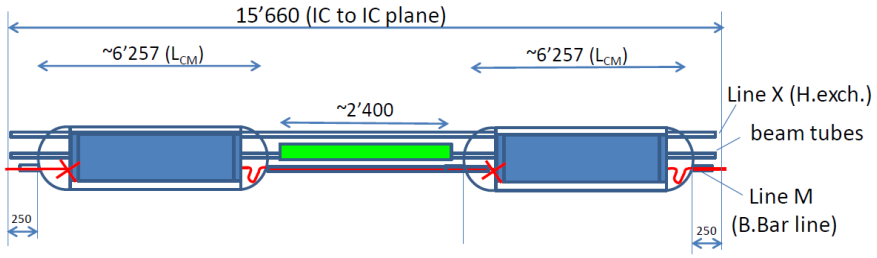

Figure 1.11: Schematic of replacement for 8 T dipoles; two 11 T dipoles with 2.4 m of space for a warm collimator. From [17].

Collimation simulations have been performed with SixTrack [18] and FLUKA [19] to compare the collimation system with and without DS collimators (called TCLDs) [8]. Results are shown in Fig. 1.12, and clearly show that TCLDs greatly reduce power deposition in SC elements from dispersive losses. Currently the space available when using the 11 T dipoles allows for an 80 cm active jaw length TCLD. As the TCLDs are not envisioned to be exposed to large beam loads, a heavy tungsten alloy is assumed sufficient for the jaw material.

1.3.4.3 Halo Diffusion

Two advanced concepts that require further R&D are crystal collimation and the hollow electron lens (HEL). Both are methods of controlling the diffusion of halo particles onto aperture limits (*i.e.* collimators). By controlling diffusion one can control loss rates or the halo population. This is useful as losses may be spread out over time, or the halo may be depleted to mitigate the effects of certain catastrophic failures.

Figure. 1.13 illustrates the use of a HEL and a crystal collimator in the LHC collimation hierarchy.

A full description of the HEL is given in Chapter 3. It has been demonstrated to provide a method of active halo control [20] at the Tevatron [5], and relies on the

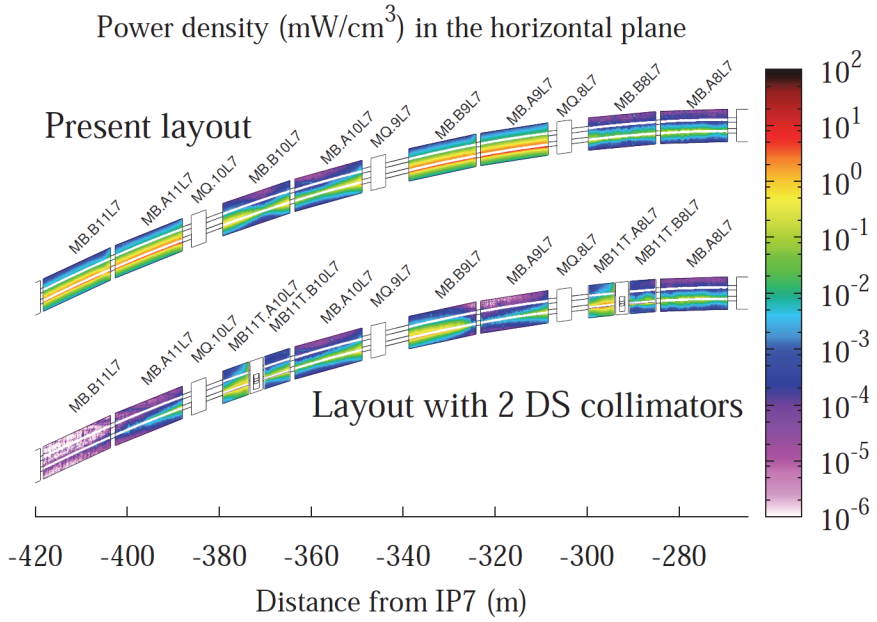

Figure 1.12: Simulated power density map in the horizontal plane of DS dipoles for nominal 7 TeV operation and a beam lifetime of 0.2 h (4.5e11 protons lost per second). Comparison of the present layout and a layout with two DS collimators. We observe the large power loads on the 11th and 9th dipole magnets (above), and the reduction after inserting collimators (below). From [8].

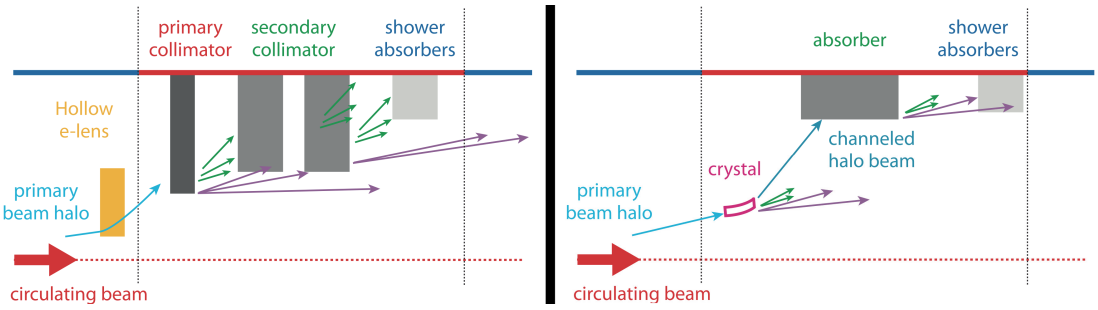

Figure 1.13: Illustration of collimation hierarchy with a HEL (left) or a crystal collimator (right). The HEL acts as a collimation enhancer, working with the current hierarchy it intercepts the halo and controls the diffusion onto the primary collimator. The crystal acts as a primary collimator, channelling the primary halo such that it may be absorbed at a much larger distance from the beam core, and thus creating a new collimation hierarchy which is dependent on the success of the crystal. From [9].

interaction of an annular beam of electrons with the LHC proton beam in order to kick halo particles such that their diffusion onto collimators is increased in a controllable

manner.

As the HEL could not be implemented until the time of a long shutdown, other methods for controlled halo diffusion were investigated in case they were needed post LS1. These are tune modulation created using quadrupoles, and excitation of halo particles using the transverse damper (ADT) system. Simulations using these methods have been coupled with machine developments (MDs), this work is still ongoing [21].

Crystal collimation for the LHC has been studied extensively over the last few years [22], however experimental tests in the LHC have only been done recently [23]. The working principle is to replace primary collimators with high purity bent crystals, which channel halo particles and steer them onto a single absorbing material. As this method of active halo control has already been studied in detail, it will not form part of this thesis.

2

Beam Dynamics

2.1 Charged Particle Motion

2.1.1 Frenet-Serret Curvilinear Co-ordinate System

For a synchrotron we adopt a co-moving co-ordinate system, as shown in Fig.2.1.

The reference orbit is a line in space through a given geometry that may be arbitrarily chosen, for example it may be taken as the closed orbit of an accelerator. The longitudinal co-ordinate s is the distance along the reference orbit from a given starting point. In the co-moving frame x and y are the transverse horizontal and vertical co-ordinates respectively, which are orthogonal to s. r denotes the radius of curvature of the reference orbit.

This co-ordinate system is known as a Frenet-Serret curvilinear system. We define this system to have units of metres, and as all three unit vectors $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{s}}$ are orthogonal, the system is orthonormal.

The two LHC beams are counter rotating, beam 1 travels clockwise, thus positive x points away from the machine centre, and beam 2 travels anticlockwise, with positive x pointing to the machine centre.

2.1.2 MERLIN Co-ordinates

We must take into account the displacement from the reference orbit in three dimensions, (x, y, ct), where ct is the longitudinal time from the reference s, normalised to units of metres. As each particle may have a spatial displacement from the reference orbit, it may also have a momentum displacement, described by (x', y', dp). The angle x' is defined as $x' = \frac{dx}{ds}$, and is unitless. y' is defined similarly. dp is the fractional offset

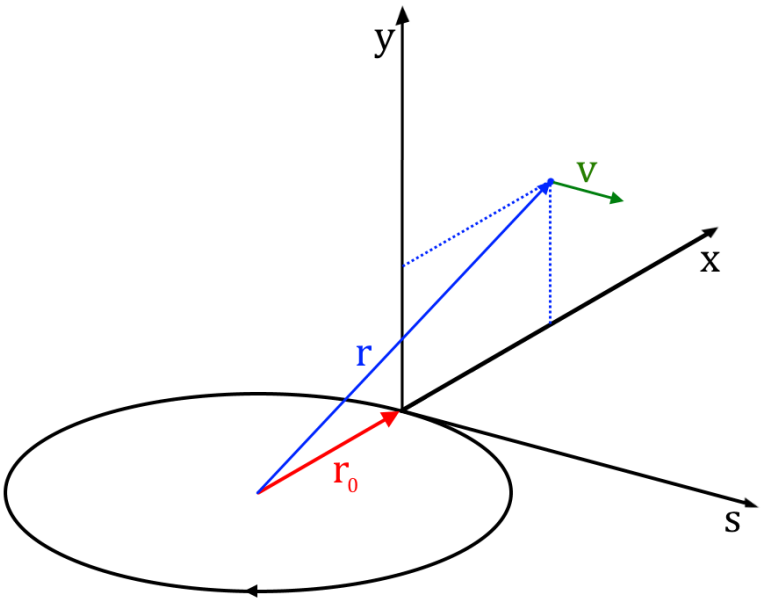

Figure 2.1: Frenet Serret co-moving co-ordinate system. The closed orbit is indicated by the black circle. Particles travel with a velocity $\mathbf{v}$ in the s direction. The transverse axes x and y indicate horizontal and vertical displacement from the reference orbit. $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{s}}$ forms the basis of the co-ordinate system. $\mathbf{r}_0$ indicates the reference orbit, and $\mathbf{r}$ the orbit of a particle.

from the reference momentum magnitude P_0, as defined in equation 2.1, and is unitless. As it will be used extensively and may be easily confused with other quantities, we will interchange dp with δ where appropriate.

$$dp = \delta = \frac{P - P_0}{P_0} = \frac{P}{P_0} - 1 \tag{2.1}$$

MERLIN uses these three co-ordinate pairs; (x, x'), (y, y'), (ct, δ) to define a particle

as a six dimensional vector $\mathbf{p}$, as shown in equation 2.2.

$$\mathbf{p} = \begin{pmatrix} x \\ x' \\ y \\ y' \\ ct \\ \delta \end{pmatrix}. \tag{2.2}$$

2.1.3 Lorentz Force

Using Newtonian mechanics we may describe the motion of a charged particle in a magnetic field by using the force, or rate of change of momentum $\frac{d\mathbf{P}}{dt} = \mathbf{F}$.

Accelerators control charged particle beams using electromagnetic fields. The Lorentz Force Law describes the force $\mathbf{F}$ exerted on a charged particle in an electromagnetic field:

$$\frac{d\mathbf{P}}{dt} = \mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}), \tag{2.3}$$

where q is the particle charge, $\mathbf{E}$ the electric field strength, $\mathbf{B}$ the magnetic flux density, and $\mathbf{v}$ the particle velocity.

Newton's second law $\mathbf{F} = m\frac{d^2x}{dt^2}$ may be used to derive the equations of motion, we may also use Lagrangian or Hamiltonian mechanics.

2.1.4 Hamiltonian Mechanics

A full treatment of the use of Hamiltonian mechanics to derive the equation of motion may be found in general accelerator physics textbooks. Here we take the approach outlined in [24].

Hamilton's equations are generally expressed as shown in equations 2.4 and 2.5;

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i}, \tag{2.4}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial x_i}, \tag{2.5}$$

where x_i are the co-ordinates of the particle, and p_i the conjugate momentum components, and $i = 1 \ldots N$ in an N-dimensional co-ordinate space. The Frenet-Serret co-ordinate system is 3 dimensional, thus we can describe the motion of a particle

in time t using 3 pairs of conjugate co-ordinates, a and p_a, where a is one of (x, y, z), by defining a Hamiltonian for the system, and using Hamiltons equations.

The Hamiltonian H defines the dynamics of the particle;

$$H = H(x_i; p_i; t). \tag{2.6}$$

For a simple mechanical system in the non-relativistic case it is expressed as the sum of the kinetic and potential energies (T and V respectively),

$$H = T + V. \tag{2.7}$$

The Hamiltonian for a relativistic particle of charge q in an electromagnetic field is given by equation 2.8 [24].

$$H = c\sqrt{(\mathbf{p} - q\mathbf{A})^2 + m^2c^2} + q\phi \tag{2.8}$$

where ϕ is the scalar potential, and $\mathbf{A}$ the vector potential of an electromagnetic field.

Thus the Hamiltonian using corresponding co-ordinates may be solved to accurately track particles through the electromagnetic fields of accelerator components.

We must perform a transformation of our independent variable from the time t to the distance along the accelerator lattice s. This is done using $\frac{ds}{dt} = v_s \frac{r}{r+x}$, and finding the derivatives of r and y, see [25] for a full derivation.

2.1.5 Transverse Motion

Using Newtonian or Hamiltonian mechanics we may derive the transverse equations of motion for a charged particle in an accelerator; Hill's Equations, which may be written as:

$$\frac{d^2x}{ds^2} + K_x(s)x = 0, \qquad \frac{d^2y}{ds^2} + K_y(s)y = 0. \tag{2.9}$$

Here K_x and K_y are periodic focussing functions dependent on the magnetic fields. A full derivation may be found in Wiedemann [26].

These second order linear partial differential equations describe a pseudo-harmonic oscillator. Particles perform 'betatron' oscillations about a stable 'closed orbit' as they travel around the synchrotron. The closed orbit is the path of a reference particle through the elements of an accelerator. Solutions to Hill's equations are not discussed

in detail here, and may be found in Lee [27].

The motion described by Hill's equations has a variable spring constant $K(s)$ which is dependent on the magnetic properties of the accelerator. As the accelerator is a synchrotron it has periodicity C, where C is the circumference of the reference particle orbit. Thus $K(s) = K(s+C)$, and we may infer that a solution has position dependent amplitude and phase. To provide a set of auxiliary functions that allow us to extract the maximum information from Hill's equations, the Courant-Snyder formalism proposes the ansatz that the solution to the transverse equation of motion is:

$$x(s) = \sqrt{\epsilon_x \beta_x(s)} cos(\psi(s) + \psi_0), \tag{2.10}$$

where β_x is a position dependent amplitude, ψ is a position dependent phase, and ϵ_x is an important constant that will be defined later. These variables are defined for each of the planes of our co-ordinate system, thus the subscript x is used for the (x, x') plane, and is interchangeable for y in the (y, y') plane. Subsequent occurrences may not be subscripted, in which case will assume that all variables in a relationship are in the same plane. We may also drop the (s) denoting that these variables are a function of the s position, in which case it is assumed that all variables are taken at the same s.

Substituting equation 2.10 into Hill's equations 2.9, we obtain two differential equations [25]:

$$\frac{1}{2}\left(\beta\frac{d^2\beta}{ds^2} - \frac{1}{2}\frac{d\beta}{ds}^2\right) - \beta^2\psi^2 + \beta^2 k = 0, \tag{2.11}$$

$$\frac{d\beta}{ds}\frac{d\psi}{ds} + \beta\frac{d^2\beta}{ds^2} = 0. \tag{2.12}$$

Integrating equation 2.12 we obtain a relationship between the phase and amplitude functions:

$$\psi(s) = \int_0^s \frac{ds}{\beta(s)}. \tag{2.13}$$

Next we define two functions $\alpha(s)$ and $\gamma(s)$ in equations 2.14 and 2.15 respectively.

$$\alpha(s) = -\frac{1}{2}\frac{d\beta(s)}{ds}. \tag{2.14}$$

$$\gamma(s) = \frac{1+\alpha^2(s)}{\beta(s)}. \tag{2.15}$$

Combining our ansatz for $x(s)$, equation 2.10, with its own derivative with respect

to s, and using α, we arrive at the definition of ϵ:

$$\epsilon_x = 2J_x = \gamma x^2 + 2\alpha x x' + \beta x'^2. \tag{2.16}$$

Where $x' = \frac{dx}{ds}$, ϵ is the Courant-Snyder invariant, sometimes called the single particle emittance (a constant of the motion of a particle), and J_x is the action variable. If we plot the Poincaré section of a particle in (x, x') phase space it will form an ellipse that has an area that is also a constant equal to $\pi\epsilon_x$.

Figure. 2.2 illustrates the relationship between the Courant-Snyder functions α, β, γ and the particle in phase space at an arbitrary s.

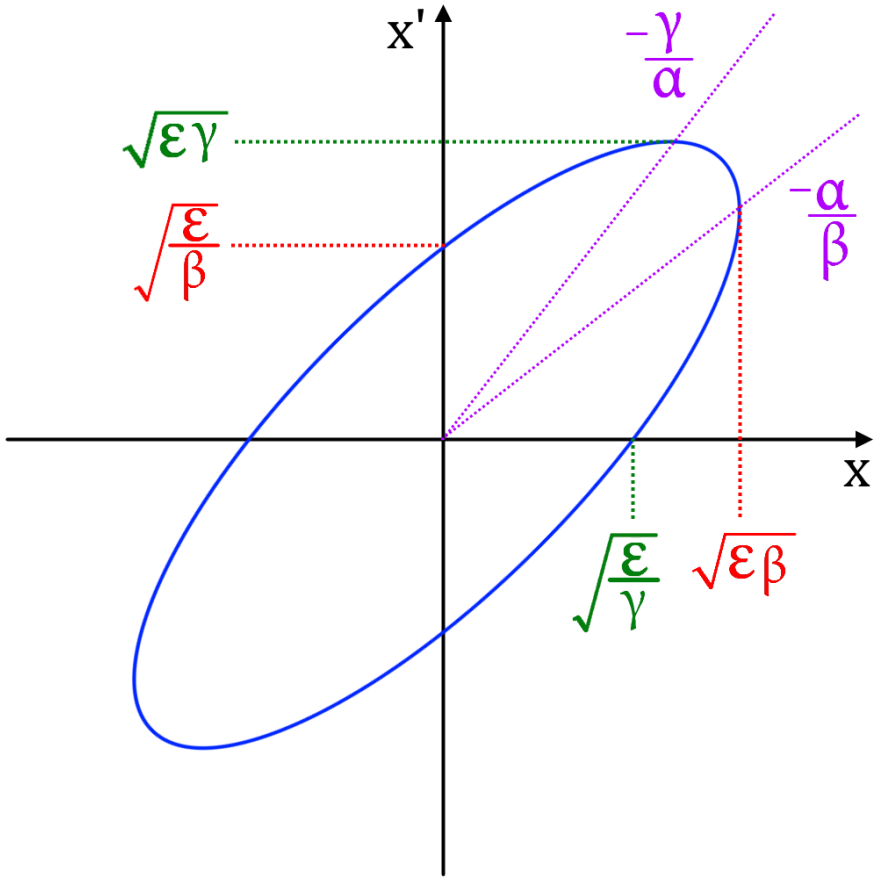

Figure 2.2: Courant-Snyder functions for a particle in (x, x') phase space at an arbitrary s. The area of the ellipse is invariant, and the shape is defined by α, β, and γ.

The β function is an important parameter, it is always positive, and is related to focussing in the accelerator. It has the same periodicity as the accelerator lattice as it

depends on the magnets. We see from Fig. 2.2 that β gives us the (phase dependent) amplitude of the particle at a given position in the accelerator. The actual size depends on the invariant also. We have defined α as half of the rate of change of the β function, thus it is maximised in focussing magnets (*i.e.* quadrupoles).

It is important to distinguish between the single particle emittance, and the beam emittance. We may define the beam emittance using the mean of the action or the particle co-ordinates in a single plane [24]:

$$\epsilon_x = \langle J_x \rangle = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle x x' \rangle^2}, \tag{2.17}$$

where ϵ_x refers to the horizontal beam emittance, and $\langle \ \ \rangle$ indicates the mean of an enclosed quantity over all particles in a bunch. ϵ will be used to denote the beam emittance from this point on.

Transfer maps may be used to compactly write the solutions of the equation of motion that takes a particle from an initial state i to a final state f. In the linear case these maps may be represented as matrices. Transfer maps must provide an explicit solution to the equations of motion, such that dynamical variables may be substituted into them.

For example the map $M(i|f)$ may be used to transform a particle defined for the vector $\mathbf{p}$ defined in equation 2.2, from an initial position s_0 to a final position s_1 in the accelerator, as:

$$\mathbf{p}_f = M(i|f)\ \mathbf{p}_i = M(s_0|s_1)\ \mathbf{p}_i. \tag{2.18}$$

As $\mathbf{p}$ has 6 dimensions, $M(s_1|s_2)$ must be a 6 x 6 matrix if linear, with further terms if higher order. In the case of linear maps, we may use the fact that matrix algebra is associative, thus a map from s_0 to s_2 may be formed from the matrices from s_0 to s_1, and s_1 to s_2:

$$M(s_0|s_2) = M(s_1|s_2)M(s_0|s_1). \tag{2.19}$$

As matrix algebra is not commutative the order of matrices must be maintained. Using this notation we define the *one-turn map*, a map that transforms a particle for a complete turn of an accelerator with circumference C:

$$M_{OT} = M(s_0|s_0 + C). \tag{2.20}$$

Thus we may represent n turns of the accelerator as $(M_{OT})^n$.

It is common to define a transfer map for each individual element of an accelerator, the contents of a transfer map M for a given element is determined by the solutions to the equations of motion in that element. We may also define a transfer matrix using the Courant-Snyder functions, also known as Twiss parameters. For one plane (x, x') the transfer matrix between s_0 and s_1 is shown in equation 2.21 [27]:

$$M(s_0|s_1) = \begin{pmatrix} \sqrt{\frac{\beta_1}{\beta_0}}(cos\psi + \alpha_0 sin\psi) & \sqrt{\beta_1\beta_0}sin\psi \\ \frac{\alpha_0-\alpha_1}{\sqrt{\beta_1\beta_0}}cos\psi - \frac{1+\alpha_1\alpha_0}{\sqrt{\beta_1\beta_0}}sin\psi & \sqrt{\frac{\beta_1}{\beta_0}}(cos\psi - \alpha_0 sin\psi) \end{pmatrix}, \tag{2.21}$$

where ψ is the s dependent phase defined in equation 2.13, and is known as the phase advance from s_0 to s_1, and the Twiss parameters are those in the corresponding (x, x') plane.

Using the periodicity of α, β, and γ we may define the one-turn map in terms of the Twiss parameters:

$$M_{OT} = \begin{pmatrix} cos\Psi + \alpha sin\Psi & \beta sin\Psi \\ -\gamma sin\Psi & cos\Psi - \alpha sin\Psi \end{pmatrix}. \tag{2.22}$$

Where the phase advance for a single turn Ψ is given by changing the limits of equation 2.13 to one turn of the accelerator, as shown in equation 2.23:

$$\Psi = \int_s^{s+L} \frac{ds}{\beta(s)}, \tag{2.23}$$

This gives us an important quantity, the machine tune ν (equation 2.24), which is a measure of beam stability and physically equates to the number of betatron oscillations of a particle in a single plane. The tune is a property of each particle, and the machine tune refers to that of the reference particle.

$$\nu = \frac{\Psi}{2\pi} = \frac{1}{2\pi}\int_s^{s+L} \frac{ds}{\beta(s)}. \tag{2.24}$$

The tune is not constant across all particles, a tune offset exists for each particle, this is dependent on the machine chromaticity ξ and the particle's momentum offset, as shown in equation 2.25:

$$\Delta\nu_x = \xi_x \cdot \delta \tag{2.25}$$

As the dispersion D arises from the difference of particle trajectories in bending

magnets due to their momentum offset δ, chromaticity ξ arises from the difference in trajectories in focussing magnets due to δ. Sextupole magnets are used to correct chromatic aberration.

If the particle tune ν is an integer, betatron oscillations will resonate and particles are lost due to machine imperfections (for example in the magnetic fields of a given magnet), thus integer tunes are not permitted in accelerators to avoid such instability. The same is true if ν is a half integer. The fractional part of ν in the transverse plane (ν_x, ν_y) may combine to cause instabilities, this occurs when the point (ν_x, ν_y) in tune space crosses a $\frac{1}{n}$ resonance, where n is an integer. Lower order resonances (e.g. $\frac{1}{2}, \frac{1}{3}$) are the most dangerous, and in most accelerators oscillations caused by high order resonances are damped on a sufficiently short time scale for them not to interfere with operation.

Although crossing such resonances causes instability in accelerators, if one has excellent knowledge of the behaviour of particles in tune space, one may purposely cross resonances in order to force losses. This may be useful for active halo control.

The trajectory of a charged particle in a dipole field is dependent on its energy. For a dipole with a purely vertical field, there is a horizontal-longitudinal coupling. We define the dispersion $D(s)$ in terms of the change in the periodic trajectory with respect to the momentum [24]. The dispersion adds an extra term to our solution to the equation of motion as shown in equation 2.26:

$$x(s) = \sqrt{\epsilon_x \beta_x(s)} cos(\psi(s) + \psi_0) + D_x(s)\delta, \tag{2.26}$$

where δ is the momentum offset defined in equation 2.1. As with the Twiss parameters, the dispersion is defined for each transverse plane.

2.1.6 Longitudinal Motion

In synchrotrons such as the LHC, particles are injected at the injection energy (i.e. longitudinal momentum), this is 450 GeV from the Super Proton Synchrotron (SPS) into the LHC. After the LHC has completed the fill of bunches, the beam is accelerated using radio frequency (RF) cavities to a top energy, this phase of operation is referred to as the ramp. The bending magnets must also ramp up synchronously with the acceleration so that particles do not deviate from a stable orbit. After reaching the top energy the synchrotron operates in 'storage ring' mode to maintain a constant orbit. The voltage of the RF cavity is a sinusoid:

$$V(t) = V_0 \; sin(\omega_{RF} \; t) = V_0 \; sin(\Phi_s). \tag{2.27}$$

The RF frequency ω_{RF} is chosen to be an integer multiple of the revolution frequency of the accelerator so that an 'on-energy' (or synchronous) particle always witnesses the same RF voltage. V_0 is the peak accelerating potential. We define $\Phi_s = \omega_{RF} t$ to be the synchronous phase, which is equal to zero when no acceleration is taking place.

There is a spread in particle arrival time due to the length of each bunch. Each particle enters the RF cavity at some phase ϕ with respect to the RF frequency Φ_s. We may write equation 2.27 for a particle with a phase ϕ as:

$$V(t) = V_0 \; sin(\Phi_s + \phi(t)) \tag{2.28}$$

For the special case of $\phi = 0$ the particle will arrive at the synchronous phase, this is the case for the reference particle.

Let us consider the storage ring case where the beam is not accelerated, instead the RF is used to maintain the beam energy. In reality particles that arrive late have $\phi > 0$, and those that are early have $\phi < 0$, as illustrated in Fig. 2.3. Late particles are accelerated, and early particles are decelerated, thus the RF cavities kick the particles towards the synchronous phase Φ_s which maintains the beam energy.

When the LHC enters the 'ramp' stage, particles are accelerated. In this case Φ_s is set to a value in the range $0 < \Phi_s < \pi$, thus the synchronous particle will gain energy at every turn. In this case particles that arrive late have $\phi > \Phi_s$, and those that are early have $\phi < \Phi_s$. As a result we observe one of two consequences. The path length around the accelerator, and thus the revolution frequency will depend on the particle momentum offset δ. We recall that $\delta > 0$ for particles with larger longitudinal momentum than the reference particle.

If the revolution frequency is higher for a particle with $\delta > 0$ then that particle will arrive at the RF cavities earlier $\phi < \Phi_s$, and we have longitudinal stability for $0 < \Phi_s < \frac{\pi}{2}$. This is because higher energy particles receive a smaller kick from the RF cavity electric field, and lower energy particles ($\phi > \Phi_s$) will receive a larger kick. Thus both cases are forced towards Φ_s, this process, known as phase focussing, is illustrated in Fig. 2.4. As a consequence we observe an oscillation in δ depending on particle arrival time ct, this is known as synchrotron motion. If the revolution frequency is lower for a particle with $\delta > 0$, the stability condition changes to $\frac{\pi}{2} < \Phi_s < \pi$. The stability of particles about Φ_s is known as the principle of phase stability.

The energy change ΔE for a particle arriving in an RF cavity at a phase ϕ is given

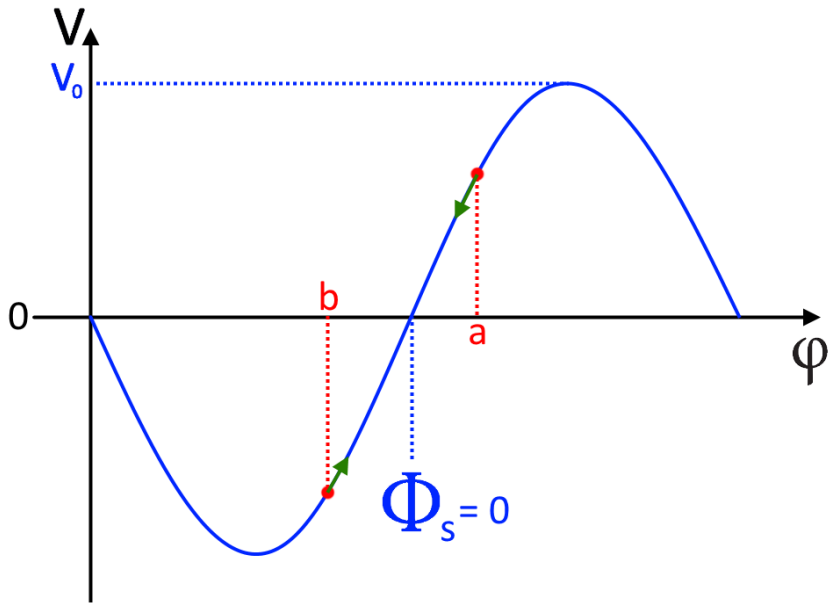

Figure 2.3: RF voltage for $\Phi_s = 0$, *i.e.* in storage mode. Particle a arrives late ($\phi > \Phi_s$) and sees a positive voltage and thus receives a positive longitudinal kick. Particle b arrives early ($\phi < \Phi_s$) and thus receives a negative longitudinal kick. Both particles are forced towards the synchronous phase Φ_0.

by equation 2.29:

$$\Delta E = qV_0 \; sin(\phi), \tag{2.29}$$

where q is the particle charge.

We can plot the Poincaré section for particles undergoing synchrotron oscillations in (ct, δ) phase space, as shown in Fig. 2.5. Here the reference particle has co-ordinates $(0, 0)$, and has $\phi = \Phi_s$. Particles with deviation in either plane perform synchrotron oscillations and circle the stable point at $(0, 0)$, however this is only the case for small deviations in both planes. The area of stability is defined by the separatrix, and is referred to as the RF bucket. As the accelerator has periodicity the beam is a train of bunches, which each occupying an RF bucket in longitudinal phase space. This means that at injection, bunches must be matched not only to the transverse optics (*i.e.* Twiss parameters), but to the RF acceptance.

Bunches of particles are thus confined to the RF bucket, particles outside the bucket

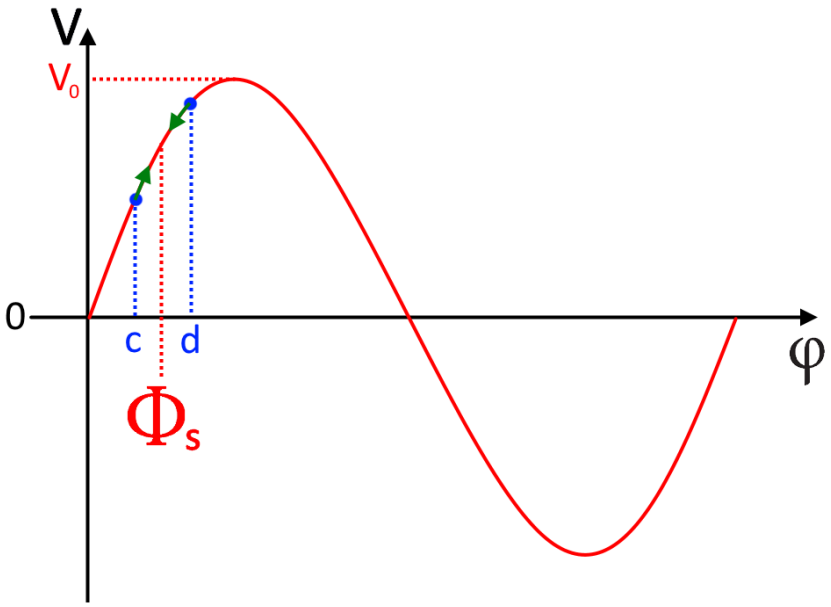

Figure 2.4: RF voltage for $0 < \Phi_s < \frac{\pi}{2}$, *i.e.* in acceleration mode. Particle d arrives late ($\phi > \Phi_s$), and particle c arrives early ($\phi < \Phi_s$). In both cases the particle received a positive longitudinal kick, however particle c will be decelerated, and particle d will be accelerated, with respect to the synchronous particle. Thus both particles are forced towards the synchronous phase Φ_0.

(referred to as the coasting beam) will inevitably be lost in an aperture of the machine, which is dangerous for machine operation.

In the LHC bunches do not occupy adjacent RF buckets, they are spaced by 10 RF buckets, which equates to 7.5 m or 25 ns at design settings. There also exists a large separation between the last and first bunch to allow enough time to trigger kicker magnets to extract the beam and transport it to the beam dump where it is deposited in a 7 m long graphite absorber encased in steel. This is the abort gap, and measures 900 m, or 3 μs.

2.2 Particle Tracking

In order to simulate the motion of charged particles in an accelerator we must take into account both transverse and longitudinal dynamics. We have defined our particle in a six-dimensional vector $\mathbf{p}$ in equation 2.2, and have established that maps may be

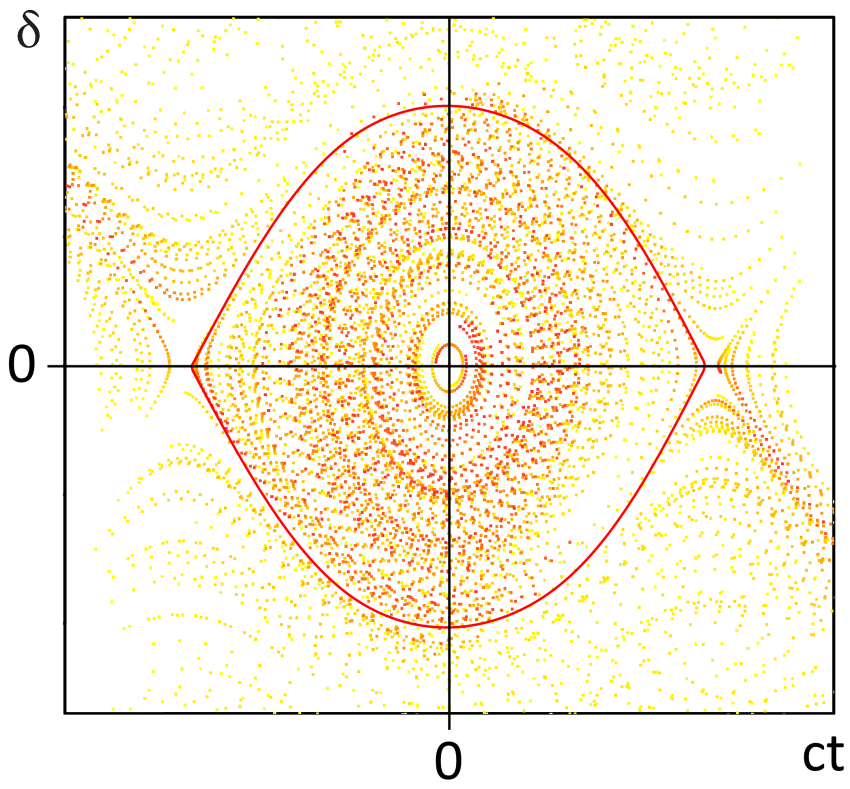

Figure 2.5: RF bucket in (ct, δ) phase space. The red line indicates the separatrix, which is the limit of stability. Particles inside the separatrix take elliptical trajectories in a clockwise direction and undergo synchrotron oscillations, particles outside the separatrix are unstable and likely to be lost after few turns.

used either in terms of magnetic field components or Twiss parameters to represent transverse particle transformations between accelerator components or for a full turn. Longitudinally we must apply the RF voltage, which is made simpler when operating the accelerator in storage ring mode (which will always be the case in this thesis).

There are various methods that can be used in order to track particles, we use the method of ray-tracing; tracking individual particle trajectories through elements of an accelerator, either element-by-element or turn-by-turn. A number of formalisms are available to minimise computational time and provide an accurate approximation of accelerator components. MERLIN offers three; `TRANSPORT`, `SYMPLECTIC`, and `THIN_LENS` tracking. Here we will consider only the first two, as they use the thick lens approximation, *i.e.* using s dependent maps for each element of length L_n, rather than the thin

lens approximation that treats each element as a thin kick at the centre of L_n, with a separation between each thin kick of $\frac{L_n}{2} + \frac{L_{n+1}}{2}$, where n is the element number.

As we are interested principally in collimation, the thick lens method allows us to split the map for any element M_i into m maps, *i.e.* $M_i = M_m \cdot M_{m-1} \cdot \cdots \cdot M_0$ (with exceptions for co-ordinate transforms at the start or end of an element). We may also use a model of the machine aperture that is continuous over s and thus perform collimation at any position in the accelerator whilst tracking with ease.

2.2.1 TRANSPORT Tracking

The TRANSPORT [28] method makes the following assumptions:

- the paraxial approximation is used *i.e.* particles have small angles x' and y',
- magnets are hard-edged and have no fringe fields,
- magnetic fields in the transverse vertical plane y are symmetric,
- linear and second order terms in the magnetic field are used, which is valid for small particle deviations around the reference orbit.

A Taylor expansion of the magnetic field $\mathbf{B}(x)$ is taken around the design orbit, and is truncated at second order. As it is not linear, tensors are required as well as matrices as shown in equation 2.30. A particle at position s_0 defined by equation 2.2 $\mathbf{p}(s_0)$ may be transformed to a position s_1 by the operation of matrices R_{ij} and tensors T_{ijk} shown in equation 2.30, and is then denoted $\mathbf{p}(s_1)$.

$$\mathbf{p}(s_1) = \sum_{j=1}^{6} R_{ij}\ \mathbf{p}_j(s_0) + \sum_{j=1}^{6}\sum_{k=1}^{6} T_{ijk}\ \mathbf{p}_j(s_0)\ \mathbf{p}_k(s_0), \tag{2.30}$$

where i refers to the elements of equation 2.2, and the integer $i = 1, \cdots, 6$.

As the TRANSPORT maps are based on a concatenated Taylor expansion of the Hamiltonian, they do not offer conservation of energy over many turns. For the creation of loss maps we observe particle losses that take only a few hundred turns, in which case this method is valid. For HEL simulations however we require thousands of turns and must use a method that conserves energy over this time-scale.

2.2.2 Symplectic Tracking

A transfer map is symplectic if it meets the condition:

$$\mathbf{J} \cdot \mathbf{S} \cdot \mathbf{J}^T = \mathbf{S}, \tag{2.31}$$

where $\mathbf{J}$ is the Jacobian of the map, $\mathbf{J}^T$ is its transpose, and $\mathbf{S}$ is the antisymmetric matrix..

The Jacobian matrix is matrix of first order partial derivatives for a vector-valued function, as defined in [24]. In the case of particle tracking the vector-valued function is the transfer map, and thus the Jacobian is a 6 x 6 matrix. The determinant of the Jacobian is denoted as $det(\mathbf{J})$. We may use matrix algebra to relate the determinants of the Jacobian $\mathbf{J}$ and the antisymmetric matrix $\mathbf{S}$:

$$det(\mathbf{JSJ}^T) = det(\mathbf{J})^2 det(\mathbf{S}) = det(\mathbf{S}), \tag{2.32}$$

since the determinant of the antisymmetric matrix $det(\mathbf{S}) = 1$ [24], it follows that $det(\mathbf{J}) = \pm 1$, and $|det(\mathbf{J})| = 1$.

Physically a symplectic map conserves the Hamiltonian, and thus the energy of the system. Approximating the solutions to the equations of motion may not satisfy the symplectic condition, however we may make certain approximations to the Hamiltonian and maintain symplecticity in the solutions derived from it.

MERLIN contains symplectic maps written by Prof. Andrzej Wolski, using the co-ordinates and method covered in [24].

As the co-ordinates used for a symplectic map are conjugate (they satisfy Hamilton's equations 2.4 and 2.5), our current phase space vector $\mathbf{p}$ is not valid. The required symplectic variables are the conjugate pairs (x, p_x), (y, p_y) in transverse phase space, where $p_{x,y}$ indicates the transverse momentum component (usually normalised to the total reference momentum p_0) in either horizontal or vertical plane. In our case we are not using the conjugate momenta but the angles x' and y'.

We will see in Chapter 4 that a change of co-ordinate definitions will impact heavily on collimation routines, and would require large amounts of code to be rewritten as well as the handling of switching between trackers etc. Instead we maintain the phase space co-ordinates of $\mathbf{p}$ as defined in equation 2.2, as the symplectic tracking in MERLIN takes into account these differences.

The co-ordinate system remains the same, and thus the positions x, y, ct are equivalent. The conjugate momenta p_x and p_y take the place of our x' and y'. As we use the paraxial approximation we assume that using $x' = \frac{dx}{ds} \approx \frac{p_x}{p_s}$, where p_s is the longitudinal momentum, is valid. y' is treated similarly. The momentum offset δ is defined the same way for both Symplectic and TRANSPORT tracking.

Thus for long term particle tracking a symplectic method is required, and now operational in MERLIN. The method was previously provided in MERLIN but due to many years of updates and added functionality, the interface between lattice definition and particle tracking required updating. Symplectic tracking will be used in this thesis for all simulations unless stated otherwise.

3

Beam Collimation

3.1 Requirements

The LHC collimation system exists as a pillar of the machine protection system (MPS). The collimation system has the following roles [9]:

- **Beam halo cleaning:** Removal of halo particles so that they do not deposit energy in other apertures which could cause; quenching of superconducting elements, radiation damage to electronics, and unnecessary activation.
- **Passive machine protection:** As the aperture restrictions of the machine, collimators provide an ongoing protection of critical elements such as the inner triplet magnets during the squeeze phase of operation.
- **Cleaning of collision debris:** Around the IPs collimators are used to protect elements downstream of the IP in both directions from debris created in IP collisions.
- **Experimental background:** By removing the beam halo, background caused by halo collision events in the IPs is reduced [29].
- **Concentration of radiation losses:** Energy deposition from lost beam particles causes activation, the collimation system ensures that this is concentrated on collimators, thus enhancing the lifetime of other elements, as well as localising radiation sources which becomes important for environmental protection and proper handling of 'hot' materials.
- **Local protection and improved lifetime:** Collimators are used to protect specific elements to ensure a longer lifetime, for example the warm magnets in IR7.
- **Halo diagnostics:** Collimator scans may be used as a diagnostic tool to measure

the beam tail population [30], this is difficult for other techniques as the bunch population is dominated by the core.

- **Halo scraping:** Collimators may be used to shape the beam, as demonstrated in special physics runs at the LHC [31].

The collimator jaw material requirements are [3, 32]:

- a good absorption rate for cleaning efficiency,
- high robustness to withstand normal and abnormal operation without damage - must withstand beam impacts of up to 500 kW of deposited energy without significant damage and maintain a jaw surface with flatness on the $10 - 25$ μm level,
- low electrical resistivity to reduce impedance,
- remote operation due to radiation levels.

These requirements are conflicting; use of a low Z (atomic number) material would imply robustness, whereas high Z materials offer better cleaning efficiency. Using metals could reduce impedance but they are susceptible to damage in accident scenarios. To meet these requirements multiple materials must be used. A multi-phased approach was adopted in the LHC, with robustness being the priority, and increased cleaning efficiency added later.

The HL-LHC requirements are similar to that of the LHC, however due to the higher luminosity beams and thus higher stored beam energy, a number of upgrades are required as described in Chapter 1. To summarise, the higher stored beam energy will result in higher loss rates on collimators and SC magnets, however the damage and quench limits remain the same as the LHC. Differences in the optics of the HL-LHC and its physical layout means that there are different and possibly new aperture bottlenecks that require local protection, and the higher luminosity gives rise to higher fluxes of physics debris for both protons and heavy ions. In addition to activation, radiation, and lifetime of hardware, the higher beam intensity also means a reduction in beam stability for similar collimator openings due to the effect of resistive wall impedance [33]. As the HL-LHC beams will have a smaller size, the corresponding collimator jaw openings will be smaller, this will also increase beam instabilities due to resistive wall impedance. New hardware such as the crab cavities (CCs) introduce new failure modes, the machine must be protected from this added risk. In particular dispersion suppressor (DS) losses

in IR7 are thought to be the limiting loss case for regular HL-LHC operation. These occur when single diffractive interactions in the IR7 collimators produce off-momentum protons with sufficient energy deviation to continue in the accelerator but be lost in the next dispersive region.

3.2 Machine Protection

The performance of the LHC as an experiment is dependent on its availability, *i.e.* the amount of time that the machine is operational. In order to take the most physics data possible, machine protection (MP) is concerned with providing the maximum beam time to experiments by anticipating possible failures, and putting in place a system to reduce the risk of such failures. In accelerators the MPS is concerned mostly with failures in handling energy and power. An uncontrolled release of the power or stored energy can result in equipment damage, unwanted activation, and loss of operation time. A particle accelerator contains many subsystems that must rely on a complex and interlocked MPS in order to operate safely, such as the RF systems, magnets, and power converters. In this section we are only concerned with the collimation aspects of the MPS, and will omit other important considerations such as energy stored in SC magnets.

Over time technology has advanced and resulted in synchrotrons with a higher stored beam energy, and smaller beam size, resulting in higher energy density and power. In this section we will summarise the major concerns of handling beams with a high stored energy in the LHC (362 MJ) and HL-LHC (675 MJ).

3.2.1 Beam Loss Monitors

The primary tools for monitoring losses in the LHC are the beam loss monitor (BLM). These detect the showers from proton impacts on an aperture restriction, and provide one of the many LHC interlocks. The system consists of around 4000 BLMS. Two types of monitor are used; ion chambers that are designed to last 30 years of operation in the LHC, and secondary emission monitors for regions with very high loss rates. The ionisation chambers are cylindrical and contain a sensitive volume of 1.5 L [34] of nitrogen at atmospheric pressure, the secondary emission monitors are similar but under vacuum and less sensitive. The BLMs must operate over a range of integration times (between 40 μs and 1.3 s [34]) to measure steady state and instantaneous losses. The BLM system provides a trigger which will dump the LHC beam if loss thresholds

are exceeded. BLMs may also be used to estimate the proton loss required to quench a superconducting magnet, for example see [35].

For a beam loss protection system the location of BLMs is determined from knowledge of possible loss locations. This requires particle tracking and loss studies, and production of simulated loss maps. This means that simulations such as those performed in this thesis, coupled with studies of secondary particle showers, can be used to determine optimal locations for BLMs in the LHC and HL-LHC. As the particle showers are in the direction of the beam, beam 1 and beam 2 losses can be separated by placing BLMs at the location of maximum secondary particle shower, usually around 1 m downstream of the loss location.

The beam interlock system operates to protect the machine and shut down the respective part of the LHC if a failure is detected. For example if a magnet quench is detected by the fast magnet current change monitors, the beam is dumped and magnet current is reduced accordingly. Other systems that can detect failures include the vacuum system, collimation system, and beam position monitor system.

3.2.2 Injection and Extraction Failures

The injection of the SPS beam into the LHC is done in batches of 288 bunches. These bunches traverse one of the transfer lines from the SPS (which includes its own collimation system), and are kicked and steered into the LHC. Errors that may occur in the SPS, the transfer lines, or the injection kickers can cause beam loss in the LHC. This type of failure is expected several times a year, and occurred twice in Run I, in one case a full batch destroyed a beam pipe in the transfer line.

The extraction of the LHC beam is performed using a system of kicker magnets which must be fired at the correct time (in the abort gap). A misfire of one of these magnets can result in parts of the beam not being extracted and being left in the accelerator possibly far from the orbit, at large transverse displacement, or outside of the RF bucket. Inevitably these particles will be lost in an aperture restriction, causing damage. This 'asynchronous beam dump' is amongst the most catastrophic expected failures, with a predicted rate of once per year, though none occurred in Run I.

3.2.3 Beam Dump

The MPS is interlocked to trigger a beam dump in case of failures that present an uncontrolled release of power or energy. Approximately 10 % of beam dumps were triggered due to beam losses in the LHC in 2012 [36].

The beam dump system is located in IR6, in which each beam-line contains 15 extraction kicker magnets (MKD) that fire in the abort gap to deflect the beam vertically into a transfer line that ends in a beam dump block. The beam is then painted onto the dump block with extraction dilution magnets (MKB) in order to dilute the energy density. The beam dump system is the only LHC system that is designed to absorb the entire beam stored energy.

3.2.4 Magnet Quenches

The LHC contains 1232 SC niobium-titanium (NbTi) dipole magnets, each 14.3 m long, operating at 8.3 T (current of 11.8 kA), and cooled to 1.9 K using super-fluid helium. The collimation system must protect these magnets from energy deposition that could result in a quench.

The SC magnets can withstand a small amount of local heating before undergoing a phase change to the normal conducting phase, this phase transition is known as quenching. A quench is dangerous for many reasons, firstly the magnetic field is no longer as desired. This may result in a deflection of the entire beam, which may cause catastrophic damage to the entire machine. A more likely scenario is that the current fed to the magnet itself will cause damage to the cryogenic systems.

The worst recorded LHC disaster occurred in a similar manner. On September 19^{th} 2008, a fault developed in the dipole bus bar in the interconnection between quadrupole Q24.R3 and the neighbouring dipole. Later discovered to be the result of faulty soldering, it caused an electrical arc as the high power could not be transported through the SC circuit (the resistance increased from $\approx 0.35\ n\Omega$ to $\approx 220\ n\Omega$). The arc punctured a super-fluid helium enclosure, and secondary arcs developed. Around 400 of the 600 MJ stored energy in the circuit was dissipated in the cold-mass and in electrical arcs. A total of 6 tonnes of helium was released, which caused a large number of 35 tonne magnets to be ripped from their anchor points, 53 of which had to be repaired. Debris contaminated the vacuum pipe for over 2 km, and the machine was out of operation for more than a year [36].

Though this failure was due to a mishandling of stored energy in the magnets, and no beam was present, it resulted in major MPS upgrades, and consolidation of all magnet splices. It limited the machine energy to 4 TeV until Run II, and necessitated 2 years of shut-down to consolidate the splices in LS1.

The maximum allowed loss rate depends on the time-scale of the loss process and energy of the particles. A constant loss of $3 \cdot 10^{-10}$ % or fast (one turn) loss of $3 \cdot 10^{-7}$

% of the nominal beam intensity is able to quench an LHC dipole magnet [37].

3.2.5 Triplet Lifetime

The inner triplet (IT) quadrupoles focus the LHC beam to a spot size of around 0.016 mm at the IPs, meaning that the beam size has a maximum of around 1.5 mm in the IT, this is illustrated in Figure 3.1. The inner triplet quadrupoles provide the limiting aperture of the LHC after collimators. As a result of their size the beams are susceptible to the non-linear fields of the magnets, thus the IT contains a number of correctors as well as the 3 main quadrupoles (Q1, Q2, Q3). The predicted lifetime of the IT is around 7 years at nominal parameters [38], or an integrated luminosity of 300 fb^{-1}. The main limitation is damage to the resin used in the inner magnet coil. the expected replacement of the IT in 2016 [39] was not required in LS1 as the LHC had a staged operation in Run I due to the 2008 disaster. As an integrated luminosity of 300 fb^{-1} is not expected to be reached until much later, a replacement of the IT is not required until LS3 (2022), coinciding with the HL-LHC upgrade.

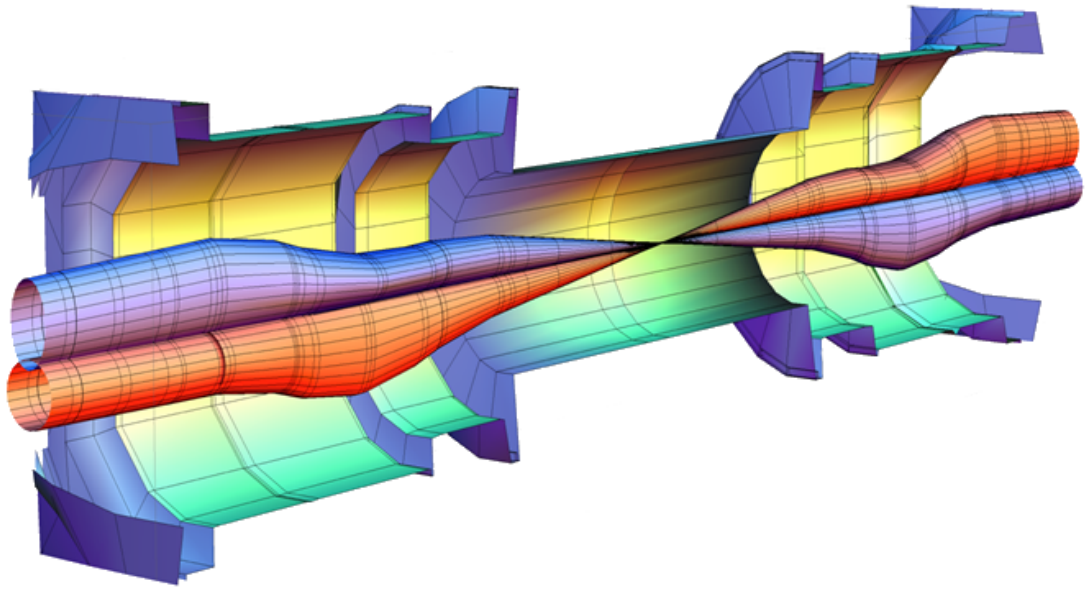

Figure 3.1: Illustration of beam envelopes in the inner triplet magnets either side of IP1 (ATLAS). The beam size is greatly reduced in the 60 m either side of the IP following the squeeze. Beam 1 is blue, beam 2 is red. From [40].

3.2.6 Crab Cavity Failures

Crab cavities are not required in the LHC, however they are for the HL-LHC. The current CC scheme for the HL-LHC includes 4 CCs pre and post IP for ATLAS and CMS, totalling 16. An earlier scheme included only 3 CCs pre and post IP, however, certain CC failures were estimated to cause a 1.5 σ orbit distortion. Measurements of LHC beam show overpopulated tails (2% outside 4σ) [41] in the transverse distribution, this would result in around 2 MJ of energy deposition in collimators [42], which is above their damage limit. The probability of such catastrophic failures resulted in the increased number of CCs in the latest HL-LHC design, as this requires a lower operational voltage which reduces the risk.

Another possible mitigation of such catastrophic damage is a method of partially depleting the beam halo, or active halo control, this could include the HEL. Highly reliable monitoring and interlocking of the halo population is also required.

3.2.7 Other Damage

Off momentum particles follow dispersive tracks in bending magnets. The dispersion suppressor regions match the optics between the long straight sections and the arcs. Due to single diffractive scattering (in the case of protons), and ultra-peripheral nuclear collisions (in the case of lead ions), dispersive trajectories have resulted in unexpected losses in the dispersion suppressor regions in IR7.

Activation of machine elements may not only reduce their operational lifetime, but mean that they are not safe for human handling. This gives rise to a small number of radiation hardened robots that perform minor maintenance on the LHC. If an element, for example a collimator, is predicted to become too active (or hot) to handle, it must be removed before this. Radiation can also effect the electronic systems used to power elements, a power supply trip can cause an emergency beam dump for example. Making electronics radiation hardened, or removing the electronics from the proximity of radiation, can help to minimise machine down time.

3.3 Losses

Losses can generally be categorised as either transient or continuous. Transient losses occur on a shorter time scale and are more catastrophic than continuous losses. Fast losses are those that cannot be handled within the response time of the beam dump system ($\approx 200\mu$s), slow losses are due to continuous processes such as the normal

diffusion of halo particles. In order to provide a rigorous machine protection system, all loss cases must be taken into account.

3.3.1 Loss Mechanisms

Particles in the same bunch can undergo multiple small-angle Coulomb scattering (MCS) as they undergo betatron and synchrotron oscillations, this is known as intra-beam scattering (IBS). Energy exchange between particles results in a coupling of horizontal, vertical, and longitudinal emittances. If these do not maintain an equilibrium, the bunch emittance continuously increases, which is the case for the LHC. The emittance growth times (the time taken for the emittance to double in a given plane) are larger than 30 hours for the LHC [32], which is longer than the operational cycle of around 24 hours.

When intra-beam scattering results in a large momentum transfer in the longitudinal plane, particles may have a large enough energy deviation to be outside the RF bucket, thus being lost or forming part of the coasting beam, this is known as Touschek scattering. The Touschek growth time is around $1.2 \cdot 10^4$ hours for the LHC [32].

Despite the ultra high vacuum (UHV) operation of the LHC (a pressure of $< 10^{-9}$ mbar), residual gas molecules may be present in the vacuum chamber. Gases such as carbon dioxide and helium are present in small amounts, and can cause nuclear scattering with beam protons. Beam gas interactions can cause protons to be kicked to large amplitudes where they are lost, and beam emittance can increase due to nuclear scattering with residual gases.

Unidentified falling objects (or UFOs) were a surprise loss mechanism in the LHC. Around 20 beam dumps were triggered in 2012 [36] because of such events. Analysis of these dumps has found that small dust particles of around 10 μm in diameter enter the beam. After this UFOs were monitored, and it was found that the vast majority of UFOs lead to losses below the dump threshold. In the injection kickers, UFOs were traced to aluminium oxide particles, subsequently a cleaning operation was performed in LS1. It was also found that the rate of UFOs was damped during operation over the year, implying conditioning due to normal operation. UFO losses can occur on a time-scale between one to tens of turns, the faster losses are below the minimum beam dump response time and can be dangerous. In order to sensitise the MPS to UFOs, a number of BLMs were relocated in LS1.

Other interactions may populate the beam halo or cause losses, for example particle interactions at the interaction points such as long range beam-beam effects. Wakefields

generated in areas of large impedance can cause bunch instabilities - the collimators themselves contribute largely to the impedance budget of the LHC.

Operational inevitabilities such as orbit drifts, optics changes, beam resonance crossing and stages such as the energy ramp can all cause beam loss and population of the beam halo. A failure of kicker magnets in either the injection or extraction system can result in a large proportion of the beam impacting upon an aperture restriction due to a translation of the beam in phase space. This can also occur due to a failure of one or more crab cavities.

3.4 Cleaning Inefficiency and Performance

Particles escaping the primary and secondary collimators may be lost in the cold aperture (SC magnets) of the machine. The global cleaning inefficiency η_c describes the performance of the collimation system, it is defined for an aperture A_c by:

$$\eta_c(A_c) = \frac{N_p(A > A_c)}{N_{abs}}, \tag{3.1}$$

where $N_p(A > A_c)$ is the number of particles leaving the cleaning insertion with a normalised amplitude $A > A_c$. N_{abs} is the total number of particles which have experienced inelastic collisions in a collimator jaw. In the LHC A_c is typically considered to be 10 σ.

It is a requirement of the LHC collimation system to operate with an $\eta_c < 10^{-3}$ at top energy, corresponding to 99.9% efficiency. A high local concentration of losses can cause quenching even if this condition is satisfied, thus we define the local cleaning efficiency $\tilde{\eta_c}$ as:

$$\tilde{\eta_c} = \frac{\eta_c}{L_{dil}}, \tag{3.2}$$

where L_{dil} is the dilution length of the losses, assumed to be $\approx 50\ m$ [32]. Using powerful simulation for tracking and aperture models, high resolution localisation of losses is possible. Loss maps provide identification of critical loss regions of the machine.

3.5 Collimator Materials

For simulations we assume the collimator is a solid block of material. The material(s) used depend on the purpose of the collimator, those currently used in the LHC are:

- INERMET® 180: by weight 95% tungsten (W), 3.5% nickel (Ni), and 1.5% copper (Cu), this material is resistant to thermal shock making it robust.
- GlidCop® AL-15: this material is mainly copper strengthened with alumina, offering similar thermal and electrical conductivities to copper, with enhanced strength.
- CFC AC-150K (carbon fibre reinforced carbon composite): composed of carbon or graphite fibre reinforcements in carbon or graphite matrices, giving it excellent resistance to corrosion and abrasion, as well as good thermal conductivity.

These materials are sufficient for the LHC requirements, however for the HL-LHC they are limited. AC-150K is limited by its electrical conductivity and radiation hardness [43], and INERMET® 180 used in tertiary collimators has low robustness in case of beam impacts [44]. It is thought that LHC performance may be limited by collimator material related concerns such as the impedance due to carbon based secondary collimators [45].

For the HL-LHC, new materials are required in order to reduce the machine impedance (the collimators are a major contributor to the impedance budget of the machine), and because the higher beam luminosity will result in higher energy deposition in normal operation and failure cases. Carbon and diamond provide low density, high thermal conductivity, and low thermal expansion, whereas metals such as copper and molybdenum provide high mechanical strength and good electrical conductivity [44]. The proposed HL-LHC collimator materials attempt to combine these materials and thus their properties. One proposed method is the use of diamonds embedded in a metallic material, two examples, copper and molybdenum diamond composites, are shown in Figure 3.2.

After preliminary investigations (see for example [47]), the following two materials are currently under investigation for HL-LHC collimator jaws:

- MoGr (molybdenum carbide graphite, or molybdenum graphite): by volume this material consists of 40% natural graphite flakes, 20% short (300 μm) carbon fibres, 20% long (3 mm) carbon fibres, and 20% molybdenum powder. Jaws made of MoGr with a Mo coating provide high robustness and reduced impedance for secondary collimators.
- CuCD (copper carbon-diamond, or copper diamond): by volume this material consists of 60% diamond, 39% copper powder, and 1% boron powder. This material is envisioned for more robust tertiary collimators for triplet protection [48].

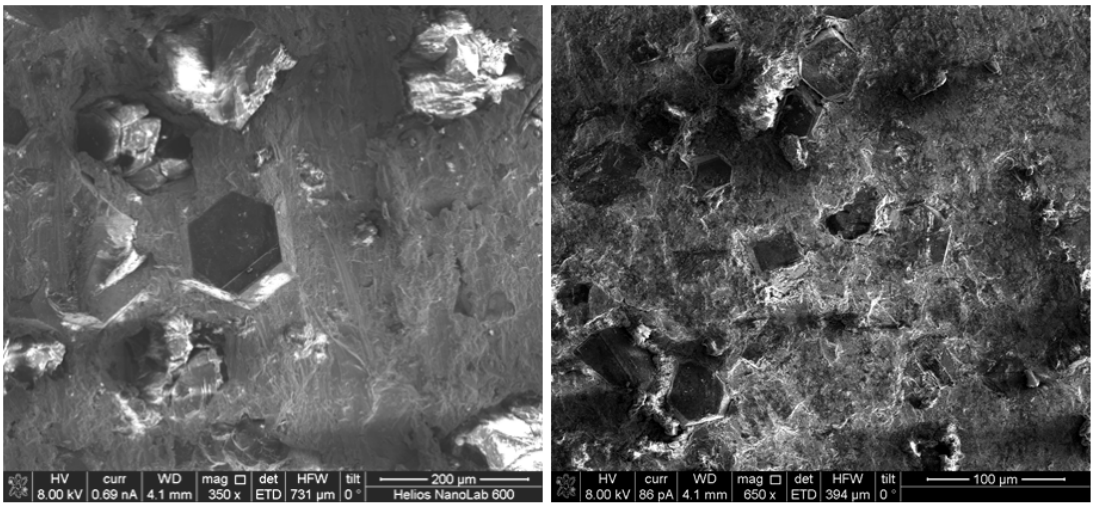

Figure 3.2: Scanning electron microscopy images of copper diamond (left) and molybdenum diamond (right) composites, showing diamond embedded in metals. The diamonds are between 45 - 100 μm wide. From [46].

Property	CuCD	MoGr	INERMET® 180	GlidCop® AL-15	CFC AC-150K
Density ρ $[\frac{g}{cm^3}]$	5.4	2.5	18	8.93	1.656
Atomic Number Z $[-]$	11.4	15.9	70.83	29	6
Thermal Conductivity λ $[\frac{W}{m\ K}]$	490	135	90.5	365	168
Melting Temperature T_{melt} $[^{\circ}C]$	1083	2505	1400	1083	3650
Electrical Conductivity σ $[\frac{MS}{m}]$	12.6	1	8.6	53.8	0.14
Young's Modulus E $[GPa]$	220	53	360	130	77
Tensile Strength $[MPa]$	70	95.7	66	413	87

Table 3.1: Comparison of existing and novel LHC collimator jaw materials. Adapted from [44, 47, 49].

A comparison of collimator jaw materials for the LHC and HL-LHC is given in Table 3.1. It is clear that both of these materials offer improved properties over current secondary and tertiary collimator jaw materials. To capitalise on the desirable electrical conductivity of metals, pure Mo or Cu coatings are also currently being investigated to reduce jaw impedance [44]. The final decision on jaw materials for the HL-LHC depends on further investigation such as radiation hardness experiments. MoGr samples have been shown to break after high doses of radiation [48], which could mean that this material is unsuitable.

3.6 LHC Collimation System

3.6.1 Multi Stage Collimation

Due to conflicting optics requirements it is necessary to use separate insertions for betatron (IR7) and momentum (IR3) collimation, *i.e.* a multi-staged collimation system. The collimation system follows a hierarchical layout in both insertions, and the full ring.

Betatron cleaning refers to the removal of particles with large betatron oscillation amplitude, i.e. the transverse halo. Thus the transverse extension of the beam halo is limited to minimise losses in SC magnets and other valuable hardware. Collimators are required to cover the full phase-space in horizontal, vertical, and skew (at an angle between horizontal and vertical) planes.

Momentum cleaning deals with the removal of particles that are off-momentum, i.e. those that would be longitudinal losses. As the horizontal dispersion is much larger than vertical dispersion in the LHC only the horizontal component need be cleaned. Momentum collimation is performed in a place of maximum dispersion in order to maximise its effectiveness.

The primary beam halo is intercepted and cleaned by primary collimators, however a secondary halo escapes. Similarly a tertiary halo escapes the secondary collimators, and in order to protect the inner triplet (IT) final focussing magnets at the IPs, a set of tertiary collimators must clean the tertiary halo, and reduce damage to the IT, as well as reducing the background to the detector experiment. Passive absorbers are also required to shield the separation and recombination dipoles D1 and D2 from physics collision debris.

3.6.2 Collimation Hierarchy

A particle impacting a collimator jaw receives a kick to both phase space co-ordinates (z, z') through the collimator length, where z is either x or y in our co-ordinates, as shown in Fig.3.3. The transverse displacement (z_{kick}) is of the same order as the impact parameter (1 μ m). The scattered protons populate an area of phase space roughly corresponding to a straight line at the amplitude of the collimator jaw.

The z' kick can cause the particles to either be scattered back into the material (or vacuum pipe further down) or into the beam. Those that return to the beam form what is referred to as the secondary halo. The secondary halo can damage the machine just as the primary, and must be controlled with the use of secondary collimators. These

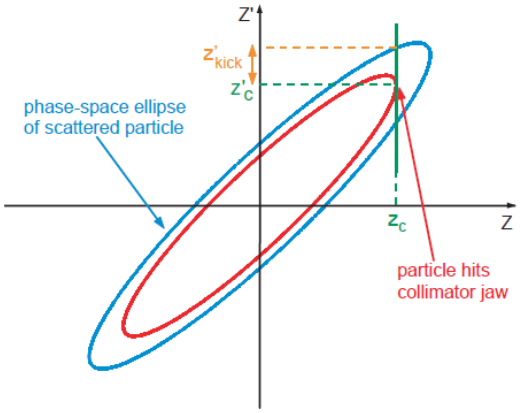

Figure 3.3: Scattering of particles in (z, z') phase space. The scattered particle achieves a larger divergence after impact (blue ellipse). From [3].

are placed downstream of the primary collimators, and the combination is referred to as a two-stage collimation system.

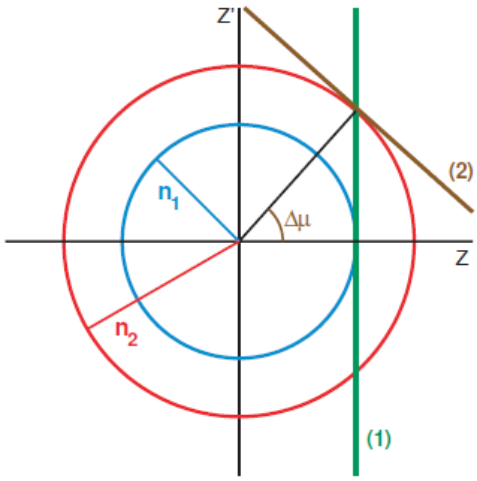

Figure 3.4: Two stage collimation system in normalised zz' phase space. A primary collimator (1) scatters a particle close to maximum transverse position $z = n_1$, if it is not absorbed it is scattered into the secondary halo (towards $z = n_2$). A secondary collimator at depth n_2 at the phase advance $\Delta\mu$ is used to control the secondary halo. From [3].

Figure 3.4 demonstrates a two stage collimation system in normalised $z - z'$ phase space. Note that the half-gap of the secondary collimators must be larger than the half-

gap of the primaries, so as to intercept only the secondary halo instead of populating it.

The secondary half-gap is set to a distance $n_2 = n_1 + d_\sigma$, where n_1 is the primary collimator half-gap, n_2 is the secondary collimator half-gap, and d_σ is the offset between the secondary and primary collimator half-gap. A 'safe' retraction d_σ of around 1σ in both transverse planes is suggested for the LHC. The downstream longitudinal distance between primary and secondary collimators is specified by a phase advance $\Delta\mu$ [50]:

$$cos(\Delta\mu) = \frac{n_1}{n_2}. \tag{3.3}$$

Figure 3.5 shows that the maximum amplitude of the scattered particles exists at two phase advances from the primary collimator (TCP). Thus two secondary collimators (TCS) are needed for each primary collimator to operate efficiently [51] as shown in Fig. 3.5. The optimum phases are also dependent on the jaw openings of primary and secondary collimators [50].

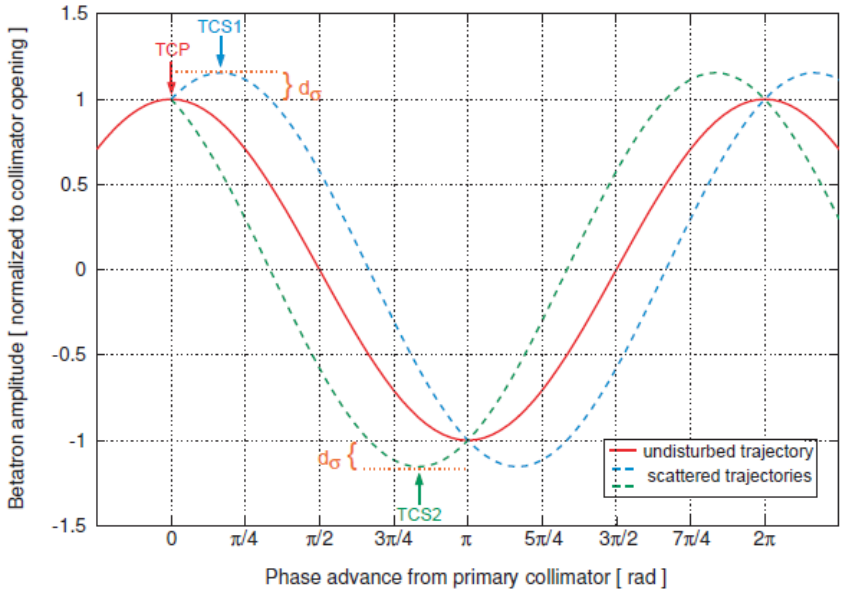

Figure 3.5: Required jaw opening of secondary collimators (TCS1) and (TCS2) with respect to primary collimator (TCP) at the optimum phase advance μ. From [51].

The LHC also makes use of Tertiary and Quaternary stage collimators to maximise efficiency. Tertiary collimators exist upstream of critical magnets in the experimental IPs, whereas quaternary collimators are mainly absorbers that protect the machine

from low energy particle showers. These are created via inelastic proton interactions with the collimator jaws. In reality the LHC collimation system has 5 stages; primary collimators, secondary collimators, absorbers in collimation insertions, tertiary collimators, and local absorbers.

3.6.3 Existing LHC Collimation System

The Phase 1 LHC collimation system prioritised robustness, it consisted of 98 two-sided and 2 one-sided movable collimators of various materials, and is shown in Fig. 3.6. Collimators must be moved as functions of time to guarantee optimum performance during energy ramp and betatron squeeze phases of LHC operation [52].

The phases of the LHC operational cycle are as follows: Injection ('setup', 'probe' or 'physics' depending on the types of beam), ramp preparation, energy ramp (where the beam is accelerated), flat top, squeeze (where the beam is focussed for higher luminosity at the IPs), adjust (when collisions are established), and stable beams (data taking periods). Collimators follow complex functions of time (four per collimator i.e. one per motor) during the ramp, squeeze and adjust modes. They sit idle at discrete settings in other modes. Three different types of interlocks (versus time, versus energy and versus β^* in the collision points) are active at all times and trigger a beam abort in case any of the collimator position or gap measurements exceed safe boundaries. The temperature of each collimator (using 5 sensors per device) is also interlocked. From the settings management point of view, this is one of the most complex LHC systems [53].

The layout of the Run I LHC collimation system is shown in Fig.3.6.

During LS1 the LHC collimation system was upgraded, as envisioned in the two-phase implementation. The goals of the upgrade were as follows [54]:

- to enhance the operation efficiency and machine protection by improving the flexibility of IR configuration,
- to improve the collimator layout in the experimental regions by providing better cleaning of incoming beam and outgoing physics products,
- to optimise the location and distribution of losses, improving the lifetime of the warm magnets and optimise doses to equipment and personnel.

This was achieved by a number of modifications. The tertiary collimators in all IRs were replaced with collimators using embedded BPMs (see Chapter 1). The primary

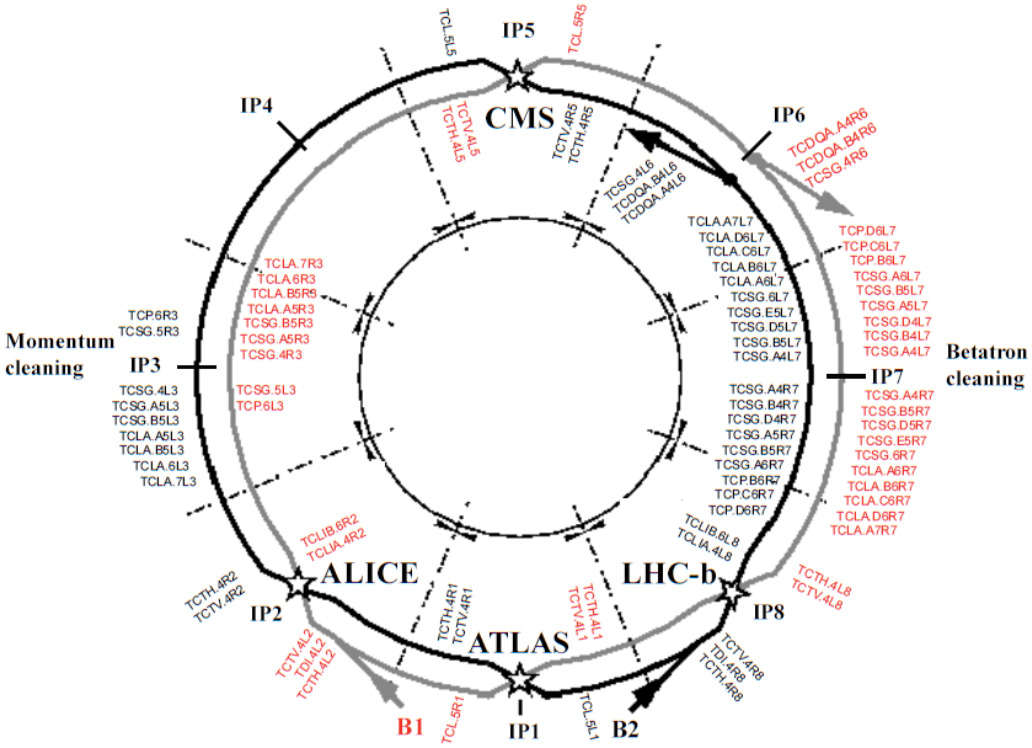

Figure 3.6: Layout of the phase I LHC collimation system for both beam 1 and 2. Red text indicates beam 1 collimators, black text indicates beam 2 collimators. From [32].

Insertion Region	Collimator Type	End of Run I	Post LS1	New in LS1
IR7:	TCP	6	6	0
Betatron	TCSG	22	22	0
Cleaning	TCLA	10	10	0
	TCSM	0	0	0
	TCLD	0	0	0
	TCAP	6	6	0
IR3:	TCP	2	2	0
Momentum	TCSG	8	8	0
Cleaning	TCLA	8	8	0
	TCSM	0	0	0
	TCLD	0	0	0
	TCAP	2	4	+2
IR6:	TCSG	2	0	-2
Beam	TCSP	0	2	+2
Dump	TCDQ	2	2	0
	TCLA	0	0	0
IR1/5:	TCTH/V	8	0	-8
High	TCTPH/V	0	8	+8
Luminosity	TCL	4	12	+8
Experiments	TCTPH/V	0	0	0
	TCLD V	0	0	0
IR2:	TCTH/V	4	0	-4
ALICE &	TCTPH/V	0	4	+4
Beam 2	TCLIA/B	2	2	0
Injection	TDI	1	1	0
	TCDD	1	1	0
	TCLD	0	0	0
IR8:	TCTH	2	0	-2
LHCb &	TCTVB	2	0	-2
Beam 2	TCTPH/V	0	4	+4
Injection	TCLIA/B	2	2	0
	TDI	1	1	0
	TCDD	0	0	0
	TCLD	0	0	0
Transfer Lines	TCDIH/V	13	13	0
Total		**108**	**118**	**+28**

Table 3.2: Summary of LHC collimators at the end of Run I, and post LS1. From [55].

dump protection collimator which consisted of two 3 m long single-sided (TCDQ) beam dump absorber blocks were replaced with three 3 m long blocks. These consist of sandwiched layers of CFC and graphite, whereas previously they were solid graphite. This particular modification was to allow safer beam dumping at higher energy beams. As well as this the layout of physics debris absorbers in the high luminosity insertions was adjusted, additional passive absorbers were added in the momentum collimation insertion, and new active absorbers were added to the dump protection collimation system. The phase 2 LHC collimation system consists of 118 collimators, of which 108 are movable. These are detailed in Table 3.2.

The name and function of each collimator type is detailed in Table. 3.3. In this acronym notation, the first T stands for target, thus TCP is a 'Target Collimator Primary'.

Both cleaning insertions (betatron in IR7, and momentum in IR3) are warm, *i.e.* have normal conducting magnets, and thus activation from halo losses is confined to these warm insertions.

Acronym	Name	Function	Material
TCP	Primary Collimator	Primary betatron cut	CFC
TCSG	Secondary Collimator Graphite (IR 3,7)	Secondary betatron cut	CFC
TCSM	Secondary Collimator Metallic	Secondary betatron cut	MoGr
TCTH	Tertiary Collimator Horizontal	Local triplet protection	W
TCTV	Tertiary Collimator Vertical	Local triplet protection	W
TCTVB	Tertiary Collimator (2-in-1)	Local triplet protection	W
TCTPH	Tertiary Collimator Horizontal + BPM	Local triplet protection	W
TCTPV	Tertiary Collimator Vertical + BPM	Local triplet protection	W
TCLA	Shower Absorber	Large amplitude shower absorber	W
TCLA	Shower Absorber (IR6)	Large amplitude shower absorber for Q4 and Q5	W
TCAP	Passive Absorber	Reduce total dose in warm magnets	W
TCL	Physics Debris Absorber	Clean matching section and DS	Cu (W)
TCDQ	Secondary Dump Protection (IR6)	One-sided dump absorption block	C
TCSG	Primary Dump Protection (IR6)	Aperture definition for dump protection	CFC
TCSP	Primary Dump Protection + BPM (IR6)	Aperture definition for dump protection	CFC
TCLIA	Injection Protection Absorber	Auxiliary injection protection	C
TCLIB	Injection Protection Absorber	Auxiliary injection protection	C
TCLD	Dispersion Suppressor Collimator	Local DS cleaning	W
TDI	Injection Protection Primary Collimator	Injection protection absorption block	C
TCDIH	Injection Protection Collimator	Transfer line injection protection	Gr
TCDIV	Injection Protection Collimator	Transfer line injection protection	Gr
TCDD	Injection Protection Mask	Movable D1 mask	C

Table 3.3: Definition of LHC collimator acronyms and corresponding materials. CFC refers to CFC AC150K, W refers to Inermet® 180, Cu refers to Glidcop® AL-15, and C is graphite. From [55].

3.6.4 Betatron Insertion

The betatron system in IR7 consists mainly of primary (TCP) and secondary (TCSG) collimators with some absorbers (TCLA). Primary and secondary collimators are two sided, and have CFC jaws, which limits energy absorption in the jaws and makes them extremely robust. To have the best compromise between cleaning efficiency and robustness an active length of 60 cm was chosen for primaries, and 1 m for secondaries. Two sided active absorbers (TCLA) intercept the tertiary halo and particle showers from the TCP and TCSGs. These use Glidcop® jaws with an Inermet® inlay to absorb maximum energy. An example betatron cleaning (IR7) system insertion is shown in Fig. 3.7. Table 3.4 lists the nominal half-gaps for the IR7 betatron cleaning system.

Collimator	**Injection half-gap** $[\sigma]$	**Collision half-gap** $[\sigma]$
TCP	5.7	6
TCSG	6.7	7
TCLA	10	10

Table 3.4: Nominal betatron collimator settings at injection and collision energy. The betatron collimation system is located in IR7. Adapted from [32].

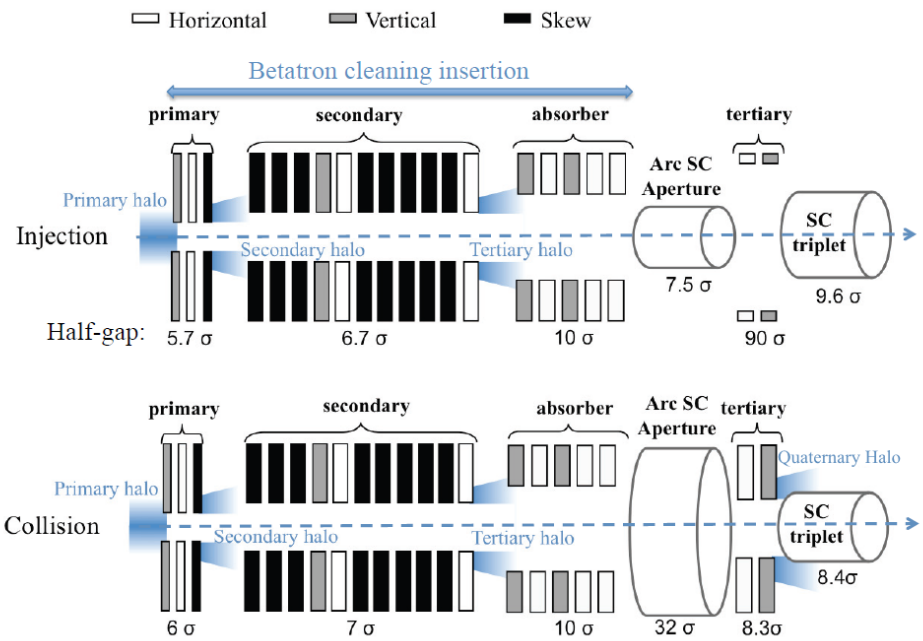

Figure 3.7: Top: 3 stage cleaning system to protect the arc cold aperture at injection. **Bottom:** At 7 TeV the machine bottleneck is given by the SC triplets rather than the arc SC magnets. Tertiary collimators are closed defining a 4-stage cleaning system. In reality another absorber stage exists after the tertiaries to define a 5 stage system. From [32].

3.6.5 Momentum Insertion

The momentum cleaning insertion in IR3 of the LHC offers an area of large dispersion. As the machine is a synchrotron vertical dispersion is negligible, thus the momentum cleaning insertion is optimized for horizontal collimation, greatly reducing the number of collimators required. This system consists of 2 horizontal primary (TCP), 16 secondary (TCS) and 4 absorbing collimators (TCLA), each of which is similar to those in IR7.

Table 3.5 lists the nominal half-gaps for the IR3 momentum cleaning system.

Collimator	Injection half-gap [σ]	Collision half-gap [σ]
TCP	8	15
TCSG	9.3	18
TCLA	10	20

Table 3.5: Nominal momentum collimator settings at injection and collision energy. The momentum collimation system is located in IR3. Adapted from [32].

3.6.6 Injection/Extraction Protection

Injection collimation systems exist in IR2 and IR8, and extraction collimation systems in IR6.

The injection protection TDI is a vertical collimator consisting of two 4.2 m long CFC jaws. Supplementary graphite TCLI collimators also provide injection error protection downstream of IP2 (beam 1) and IP8 (beam 2). Due to their half-gaps (shown in table 3.6) the TCLIs must be retracted for the energy ramp phase.

Three 3 m long extraction protection TCDQA collimators made from sandwiched layers of CFC and graphite are installed in IR6. A supplementary TSCG collimator provides further extraction error protection downstream of the TCDQAs.

Collimator	**Injection half-gap** $[\sigma]$	**Collision half-gap** $[\sigma]$
TDI (Injection)	6.8	N/A
TCLI (Injection)	6.8	N/A
TCDQA (Extraction)	8	8
TCSG (Extraction)	7	7.5

Table 3.6: Nominal injection/extraction protection collimator settings at injection and collision energy. The injection protection collimation system is located in IR2 and IR8. The extraction protection collimation system is located in IR6. Adapted from [32].

3.6.7 IP Protection

At top energy the beam must be squeezed in order to reach the nominal β^* values at the IPs. This means that the β function at the SC triplet magnets (quadrupoles Q1, Q2, and Q3) must increase, and thus the available aperture is reduced in the IPs. Horizontal TCTH and vertical TCTV tertiary collimators are installed upstream of the SC triplets to provide protection during squeeze and collision. These consist of two sided, 1 m long copper jaws with a tungsten inlay. Tertiary vertical collimators in IR1 and IR5 have the classical one-beam design - the TCTVA is identical to the TCLA. Following the squeeze of the beam, when the triplet aperture becomes smaller than the arc aperture, the TCTs must also be closed [32]. Table 3.7 lists the nominal half-gaps for the IP protection system.

The TCTs clean the tertiary halo that leaks from the collimation insertion in IR7, and may be used to tune experimental backgrounds. In the case of a catastrophic failure the TCTs are designed to intercept a large proportion of the beam energy, thus protecting the SC triplet quadrupoles, however the TCTs are likely to be destroyed in

Collimator	Injection half-gap $[\sigma]$	Collision half-gap $[\sigma]$
TCTH/V (IR1-5)	9.5	8.3
TCTH/V (IR2)	9.2	45
TCTH/V (IR8)	9.2	30

Table 3.7: Nominal IP protection collimator settings at injection and collision energy. Adapted from [32].

such a failure.

As well as active TCTs and absorbers, the high luminosity IPs (1 and 5) contain two passive absorbers on each side of the IP. The TAS (target absorber) protects the inner triplet from collision debris. The TAN (target absorber neutral) protects the beam separation dipole (D2) from neutral products of collisions in the detector region. A full layout of IP1 is shown in Fig. 3.8.

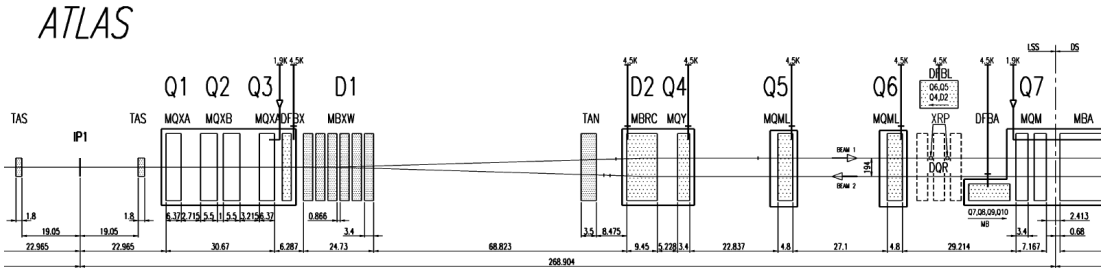

Figure 3.8: Layout of the ATLAS interaction point (IP1). IP1 shows the collision point, followed by the TAS, the inner triplet (Q1, Q2, Q3), the separation dipole D1, the TAN, the recombination dipole D2, and subsequent quadrupoles (Q4 - Q7). From [56].

3.6.8 Run I

A more up to date table of collimators used before LS1 is shown in table 3.8

The LHC ran successfully between 2008 and 2013 operating at energies of up to 4 TeV per beam. This equates to 140 MJ stored beam energy at a peak luminosity of $7 \cdot 10^{33} cm^{-2} s^{-1}$ [53].

Losses in the post collimation and post interaction point dispersion suppressor regions were observed, which were later identified as those from off-momentum protons having undergone single diffractive interactions in the collimators. For ions, similar losses were observed due to bound-free pair production [53]. No magnet quenches were observed from circulating beam loss.

Functional Type	Name	Plane	Number	Material
Primary IR3	TCP	H	2	CFC
Secondary IR3	TCSG	H	8	CFC
Absorbers IR3	TCLA	H, V	8	W
Primary IR7	TCP	H, V, S	6	CFC
Secondary IR7	TCSG	H, V, S	22	CFC
Absorbers IR7	TCLA	H,V	10	W
Tertiary IR1/2/5/8	TCT	H, V	16	W/Cu
Physics abs. IR1/5	TCL	H	4	Cu
Dump protection IR6	TCSG	H	2	CFC
	TCDQ	H	2	C
Inj. protection (lines)	TCDI	H, V	13	CFC
Inj. protection IR2/8	TDI	V	2	C
	TCLI	V	4	CFC
	TCDD	V	1	CFC

Table 3.8: LHC collimators used in the 2010-2013 run. CFC refers to CFC AC150K, W refers to Inermet 180, Cu refers to Glidcop AL-15, and C is graphite. From [53].

3.6.9 Run II

To prepare for the increase of beam energy from 4 TeV to 6.5 TeV per beam, 12 TCL collimators were added to the LHC to protect from collision debris downstream of ATLAS (IP1) amd CMS (IP5). These additional collimators improved the flexibility for forward physics experiments and allowed concentration of radiation hot spots [57]. Validation loss maps were generated (using the transverse damper to excite losses and measuring losses using BLMs), and showed the highest cold losses to be well below the required cleaning inefficiency (≈ 0.0001) [57]. As in the previous run the largest cold losses were found to be in the dispersion suppressor magnets post betatron collimation (IR7). Due to the excellent performance of the collimation system, the $\beta*$ was pushed to $\approx 30\%$ beyond its nominal value [57]. By adding collimators with embedded BPMs in LS1, alignment of the collimators was performed in a matter of seconds (as compared to hours in the previous run), and the stability of the collimation system (variation in cleaning inefficiency over time) was much improved.

3.7 HL-LHC Collimation System

3.7.1 Possible Upgrades

The current HL-LHC baseline includes 56 new collimators (some added and some replaced), to be installed in LS3 [48].

The HL-LHC optics present a new aperture bottleneck at the high luminosity IPs (IP1 and IP5), presently a pair of horizontal and vertical TCTs protect the inner triplet quadrupoles (Q1, Q2, Q3) from the beam halo, and provide passive protection in case of an asynchronous beam dump or equivalent failure. The potential bottleneck occurs upstream of the TCTs, at the Q4 and Q5 quadrupoles, thus an additional set of TCTs is required to protect these quadrupoles [58]. This results in 4 additional 'TCT5's per beam, two in IP1, two in IP5. In addition these IPs will require additional physics debris absorbers, and fixed masks at each of the three absorbers [33].

To reduce losses in the dispersion suppressor regions on either side of the betatron cleaning insertion (and at other areas where this is an issue), DS collimators (TCLD) are required. Two TCLDs are required per beam in IR7, with at least one per beam in IR2 due to ion losses from collisions in ALICE. Half of these are planned to be installed in LS2, the IR7 TCLDs must be placed between 11 T dipoles (as discussed in the next section), but the IR2 TCLDs do not require the high gradient dipoles. The rest of the TCLDs and corresponding 11 T dipoles will be installed in LS3 if they are deemed necessary.

The LHC is limited by the robustness and impedance of the collimators. As mentioned previously a large research and development campaign was started in order to identify and test possible materials for collimator jaw upgrades (for the LHC and HL-LHC). The current TCSG material (CFC) results in an impedance limitation to the HL-LHC, and a staged replacement with MoGr (or suitable alternative), possibly with a metal coating, is envisioned for the HL-LHC. In total 22 IR7 secondary collimators require replacement, 8 - 10 of these will be completed in LS2 [48], with the IR3 collimators being replaced if deemed necessary.

The TAN is a passive neutral absorber that is required in IP1 and IP5. It protects the twin-aperture SC separation dipoles (D2) from forward high energy neutral products from collisions, and localises activation. In the LHC each of these IPs require two TANs, each absorbing around 210 W of neutral particles when the LHC is colliding beams [59]. For the HL-LHC an upgraded TAN (named TAXN) is being designed [60].

Energy deposition studies have indicated that a large leakage occurs if using the

present TCL physics debris absorber jaws. This has prompted an updated TCLX design with larger jaws to protect magnets from D2 to Q4 [60]. The TCL4 tertiary collimator has jaws that have a transverse cross section of 2 × 3.4 cm, the TCLX instead uses 4 × 7 cm.

As well as these upgrades are the novel collimation schemes such as crystal collimation, hollow electron lens collimation, and the use of a rotary collimator, all of which are discussed in Chapter 1.

3.7.1.1 Dispersion Suppressor Collimator

FLUKA simulations have indicated a power deposition of > 90 mW cm^{-3} in the SC magnets in the DS region post IR7 due to heavy ion losses in the HL-LHC. This is higher than the quench limit of ≈ 50 mW cm^{-3} [61].

In order to alleviate such losses a collimator is required in the DS region post IR7. The losses occur in a region currently occupied by SC dipoles, thus there is no space for a collimator. Two options exist; shifting many magnets up or down-stream in order to fit a collimator into the lattice, or replacing an 8 T dipole with two higher field magnets, thus making space for a collimator between. The latter was chosen, and a prototype 11 T dipole was designed and built at CERN.

The DS is a cold region of the machine, and preliminary studies showed that a cold collimator (*i.e.* a collimator in a cryostat) was not practical [62]. Thus a warm collimator with a cryo-bypass that sits between two cold 11 T SC dipoles is envisioned for the HL-LHC. The current design is shown in Fig. 3.9.

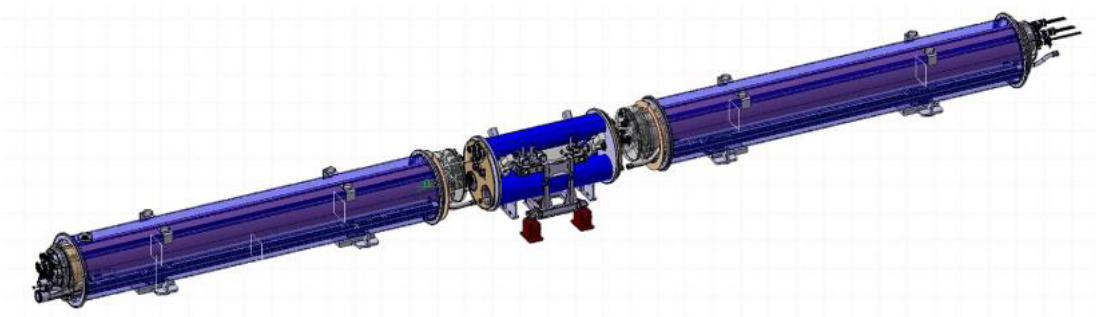

Figure 3.9: Proposed design of two 11 T dipoles with a dispersion suppressor collimator (TCLD) collimator between them. This design would replace an existing 8 T dipole. From [62].

SixTrack simulations indicate a factor 10 reduction in peak power deposited in the SC magnets in the DS regions when operating with two TCLDs in IR7 for a single

beam [33].

3.7.1.2 Novel Collimation Schemes

Crystal collimation offers improved cleaning performance and impedance over current primary collimators. By using a bent crystal to channel halo particles onto a single absorber, the current multi staged collimation approach may be bypassed. Machine development (MD) studies were carried out in 2015 demonstrating crystal channelling in the LHC (6.5 TeV protons and 450 GeV lead ions). Cleaning at top energy has yet to be demonstrated, as well as channelling for 6.5 TeV ions. A further concern is the performance of crystal collimators in dynamic machine phases such as the ramp and squeeze.

Hollow electron lens collimation has not yet reached the stage of crystal collimation. The current design includes a 5 T, 3 m long SC solenoid (in parts) in a cryostat, with bending toroid resistive coils, in an S shaped layout with electron gun and collector at the extremities [63]. The HEL solenoid is superconducting and operates at 4.2 K, thus access to liquid helium is necessary. The HEL design is shown in Figure 3.10.

The HEL requires compact construction, maximum vacuum chamber diameter, easy access for installation, modification and maintenance, and further insertions such as BPMs and diagnostics in order to align the HEL beam with the proton beam. It operates on a single beam, so a large beam to beam distance is required. Two candidate locations in IR4 have been suggested; RB44 and RB46. The beam to beam distance at these locations is 420 mm.

Another interesting and novel scheme for the HL-LHC is to compensate beam-beam effects using an electrical wire embedded inside a TCTP tertiary collimator jaw near the IPs. A high DC current of 350 A flows through the wire which is positioned less than 3 mm from the jaw surface, whilst maintaining TCT functionality. No change in damage limits from normal Inermet® TCTs has been found, and four collimators with wires for beam-beam compensation are included in the HL-LHC baseline.

In order to correctly set the collimator positions and jaw openings, a very accurate knowledge of the beam orbit and size is required. One cannot rely on the nominal values for these variables as imperfections and errors cause uncertainties that are large when compared to the collimator gaps. Thus deterministically setting the collimators is not an option, and instead a beam-based alignment technique is used. This was done manually at the start of Run I, taking around 20 minutes per collimator. The optimisation of beam-based alignment was performed and reduced this set up time to

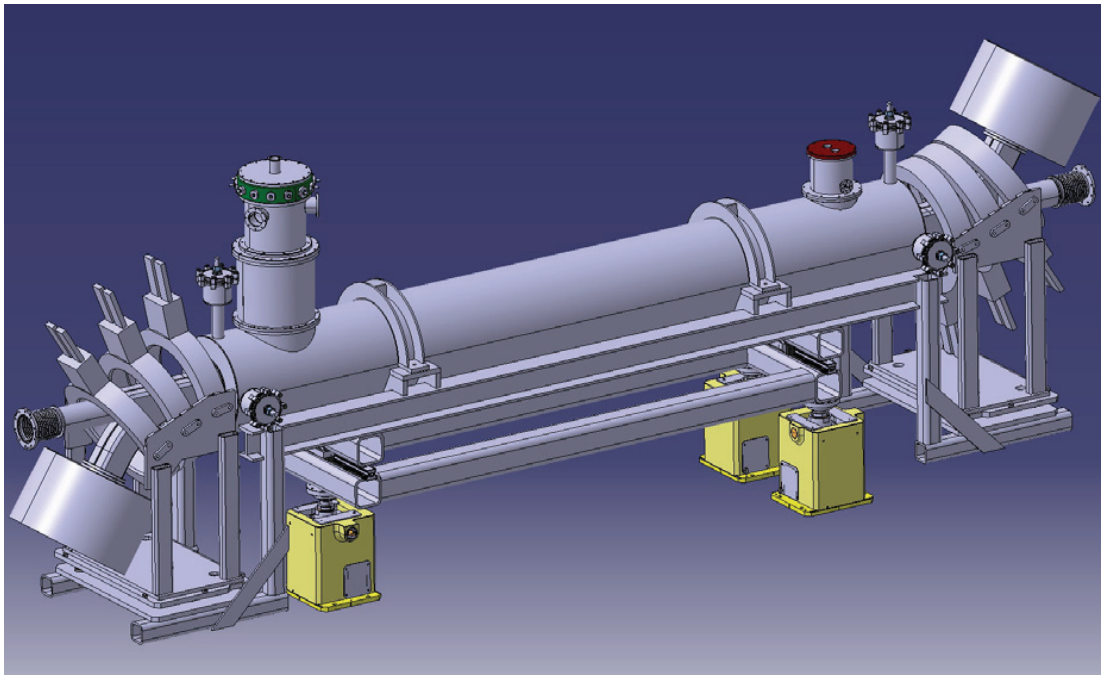

Figure 3.10: Proposed design of an 'S-shaped' LHC hollow electron lens. The electron gun and collector are located in the extremities of the 'S', and the injection and extraction toroids are shown in the bends. The main section is a SC solenoid, and the beam pipe for protons extrude on either end. From [63].

less than 5 minutes per collimator [4]. BPMs were installed in a number of collimators in LS1, and an upgrade of all collimators to include these BPM 'buttons' is foreseen in the HL-LHC, in order to provide better halo diagnostics and reduce collimator set up time. Collimators with embedded BPM buttons were installed in LS1, and are shown in Figure 1.10.

3.7.2 Proposed Layout

The proposed layout of the HL-LHC, with the addition of one HEL per beam (not currently in the HL-LHC baseline), is shown in Fig. 3.11.

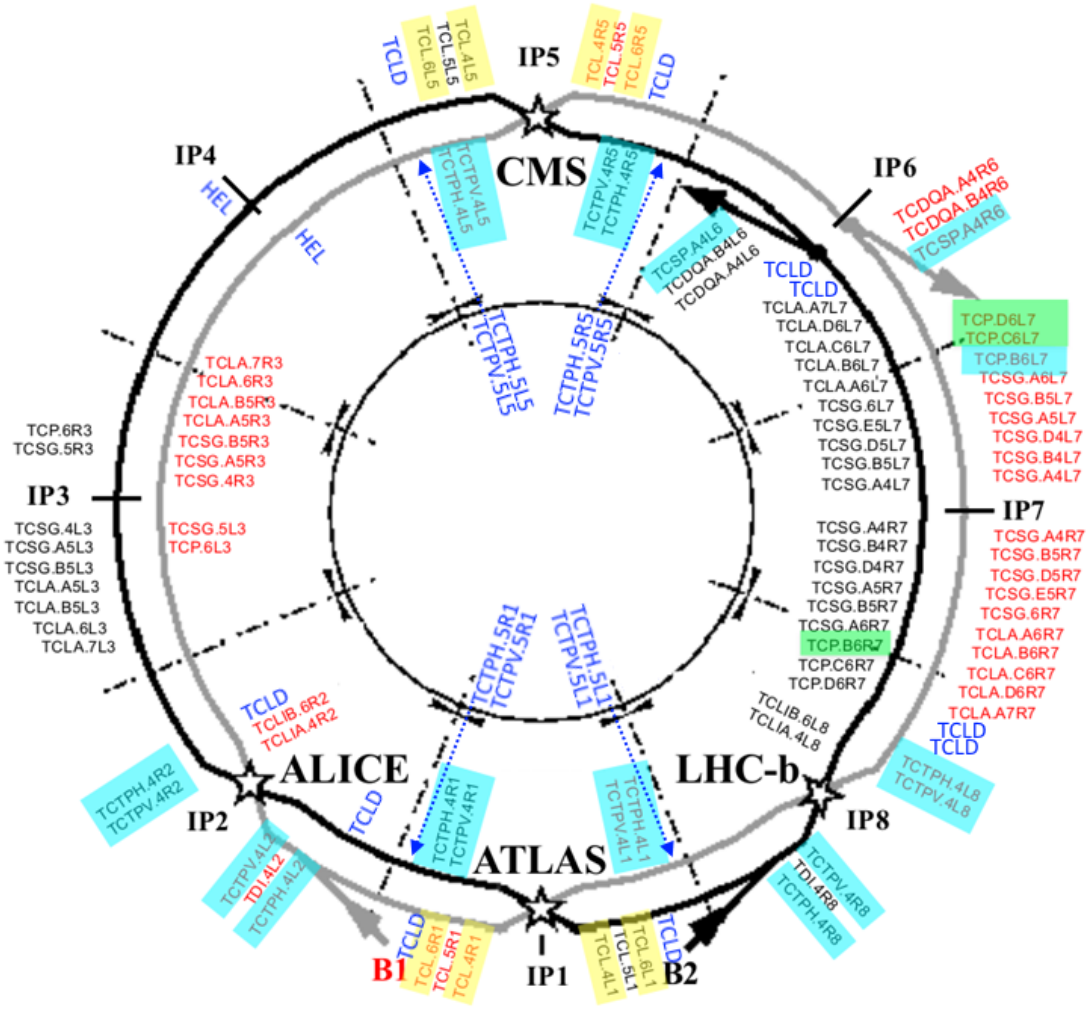

Figure 3.11: Layout of the LHC collimation system after LS1 for both beam 1 and 2, with possible additions for HL-LHC, and two HELs (which are not currently part of the baseline but give an indication of their possible positions). Yellow highlights indicate newly installed collimators in LS1, blue highlights indicate collimators that were replaced (for example with embedded jaw BPMS), and green highlights indicate collimators that were removed and put back in place. Red text indicates existing beam 1 collimators, black text indicates existing beam 2 collimators, and blue text indicates possible HL-LHC upgrades. Adapted from [33].

3.8 Hollow Electron Lens Collimation

The hollow electron lens (HEL) is an annular beam of electrons which may be used to interact with the LHC proton halo in order to control halo diffusion, thus providing a method of active halo control.

3.8.1 Tevatron Operation

The HEL was demonstrated for the first time at the Tevatron [20] p^+p^- collider. The Tevatron Electron Lens (TEL) was used was used for abort gap cleaning [64] 24/7 for a duration of five years with only a few eight hour accesses to replace failed components [65], thus demonstrating compatibility with regular collider operation. TEL used a solid electron beam. The second lens, TEL-2, was used successfully for long range beam-beam compensation over a period of months [66].

TEL-2 was the first hollow electron lens to be used in a hadron collider, it demonstrated controllable halo particle removal without affecting the beam core [20]. The applications of electron lenses are summarised in [67].

3.8.2 Existing Operation Modes

Four existing operation modes exist for collimation enhancement using the HEL, these are detailed in [67], and summarised below:

- DC: The HEL current is continuously at maximum. Halo particles receive a kick that is proportional to the particle transverse displacement (as in equation 3.7), and collimation enhancement relies on the coupling of this kick with machine resonances.
- AC: The HEL current is modulated over time in order to drive the betatron oscillations of halo protons. Collimation enhancement is achieved by increasing transverse displacement of the forced betatron oscillation.
- Diffusive: The HEL is either switched on/off, or the current is modulated, on a turn by turn basis. It is not possible to do this on a bunch by bunch basis in the LHC. By applying this randomly modulated kick to the halo protons, the natural diffusion of the halo, and thus the collimation enhancement is performed in a controlled manner.
- Turnskip: The HEL is switched to DC mode every n turns, in an attempt to drive the betatron oscillations as in the AC mode.

3.8.3 LHC HEL

In the HL-LHC beam sizes are smaller than in the LHC, requiring smaller collimator gaps and thus larger impedance contributions, and corresponding beam instability. The HEL can operate closer to the beam core than a collimator jaw without contributing significantly to the impedance budget, this may also allow some collimators to have larger jaw openings, as the HEL will kick halo protons onto the primary collimators to deplete the halo. As the HEL is not solid there is no material to damage, and due to the SC solenoid the HEL beam is very well controlled. In controlling halo depletion the HEL offers a method of reducing the damage caused by catastrophic failures. If the halo is depleted before such a failure, the intensity of protons that would impact upon the collimators or magnets would be decreased. This is especially useful for the HL-LHC due to the use of crab cavities, which are a major concern in terms of possible failures.

The use of a HEL for LHC collimation was first suggested in 2006 [65], in which the highly reliable Tevatron hardware was summarised and scaling of this for use in the LHC was suggested. It was stated that for the LHC two to three HELs per beam, each operating with current of 10 A, length of 2 m, and voltage of 20 kV would be necessary to obtain controlled diffusion of the halo similar to that achieved in the Tevatron.

An investigation was performed in 2013 [68], in which SixTrack [18], the standard CERN collimation simulation tool was modified and used to perform simulations of a HEL in the nominal LHC lattice using hardware similar to the TEL. It was found that; the AC mode was most effective at exciting halo particles onto the primary collimator (around 75% removal in $\approx 20\ s$) after the AC parameters were optimised, the DC mode showed no noticeable effect, and the diffusive mode was less effective than the AC mode. It was also observed that doubling the current in diffusive mode was comparable to the AC mode removal rate.

Later in 2013 the HEL was mentioned in the HL-LHC preliminary design report [69], though not in the HL-LHC baseline at that time it was highlighted as a possible means for collimation enhancement in the HL-LHC.

After further study a conceptual design report (CDR) was produced in 2014 [70], in which experimental experience from TEL operation and numerical simulations were used to produce a conceptual design for a HEL that met the requirements for HL-LHC collimation enhancement. The hardware parameters of the TEL and LHC HEL are summarised in Table 3.9.

At the proposed location for the HEL in the HL-LHC, the beam is not round. The

Parameter	Tevatron	LHC
Interaction Length $[m]$	3	3
e^- Energy $[keV]$	5	10
e^- Current $[A]$	1.2	5

Table 3.9: HEL hardware properties

HEL beam is round due to the cathode shape, and is maintained by the magnets and space charge forces [67]. Collimation enhancement is thus optimal in regions where the proton beam is round. We therefore compare the position at RB46 (for beam 1) where the beam is non-round, to another position nearby where the beam is round, in order to compare the effect on collimation enhancement.

For this study the parameters detailed in Table 3.10 were used. For SixTrack comparisons, the Tevatron HEL parameters shown in Table 3.9 were used, with LHC optics version 6.503, shown in the LHC_{Tev} column of Table 3.10.

The remaining columns of Table 3.10 compare the round and non-round positions for HL-LHC optics version 1.2, with collision or separation. It is clear that the difference in optics is negligible. This study therefore uses only collision optics.

Parameter	$\mathbf{LHC}_{Tev}$	$\mathbf{HL}_{Sep}$ Round	$\mathbf{HL}_{Sep}$ Non-round	$\mathbf{HL}_{Coll}$ Round	$\mathbf{HL}_{Coll}$ Non-round
s $[m]$	10032	9967	9908.4	9967	9908.4
β_x $[m]$	181.7	211.1	331.7	213.4	331.7
β_y $[m]$	180.4	217.2	211.9	215.9	211.9
α_x	-0.32	-0.8	-1.26	-0.82	-1.26
α_y	0.96	0.18	-0.09	0.18	-0.09
D_x $[m]$	-0.14	-0.31	-0.43	-0.35	-0.5
D_y $[m]$	0.08	0.15	0.09	0.15	0.08
σ_x $[\mu m]$	292	315	395	316	395
σ_y $[\mu m]$	291	319	315	318	315
$\sigma_{x'}$ $[\mu rad]$	1.67	1.9	1.9	1.91	1.91
$\sigma_{y'}$ $[\mu rad]$	2.24	1.5	1.5	1.5	1.5
μ_x	25.16	24.34	24.25	24.35	24.25
μ_y	22.85	22.68	22.67	22.7	22.67

Table 3.10: HEL simulation lattice functions for nominal LHC (v6.503) and HL-LHC (v1.2) optics. Separated beam (*Sep*) and collision (*Coll*) optics are shown for both round and non-round positions in the HL-LHC.

3.8.4 HEL Modelling

The machine beam interacts with the electromagnetic (EM) field generated by the HEL beam, the HEL interaction is not based on scattering [67].

3.8.4.1 Assumptions

In order to derive the force on a proton interacting with the HEL EM field, we assume a perfect HEL:

- The HEL is a perfect hollow cylinder, with no variation of thickness, or inner or outer radii (R_{min} and R_{max} respectively) through its interacting length L.
- The electron distribution is perfectly uniform both longitudinally and radially.
- The electron charge density is azimuthally symmetric.

These assumptions mean that any proton inside the HEL minimum radius R_{min}, i.e. the beam core, feel no net force and is therefore not affected by the HEL. This ideal HEL is shown in Figure 3.12.

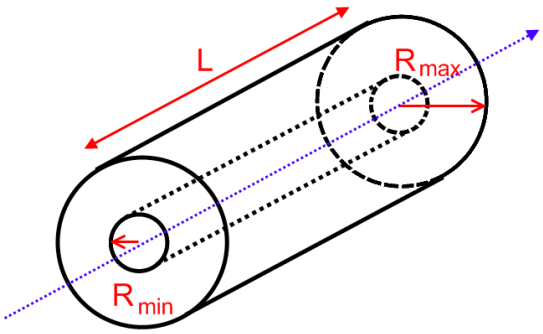

Figure 3.12: Diagram of a perfect HEL beam, the blue line indicates the proton beam axis L is the active length, R_{min} the minimum radius, and R_{max} the maximum radius.

For modelling we also ignore the effect on protons from the magnetic fields, *i.e.* the SC solenoid, and the injection and extraction toroids. The HEL beam can overlap the proton beam at its injection and extraction points, we ignore this effect as it was demonstrated not to affect beam intensity or emittance at the Tevatron [67], and we assume the same to be true in the LHC case.

3.8.4.2 HEL Kick

The full kick derivation may be found in Appendix B. We model the HEL beam as an infinite line charge, as shown in Figure 3.13.

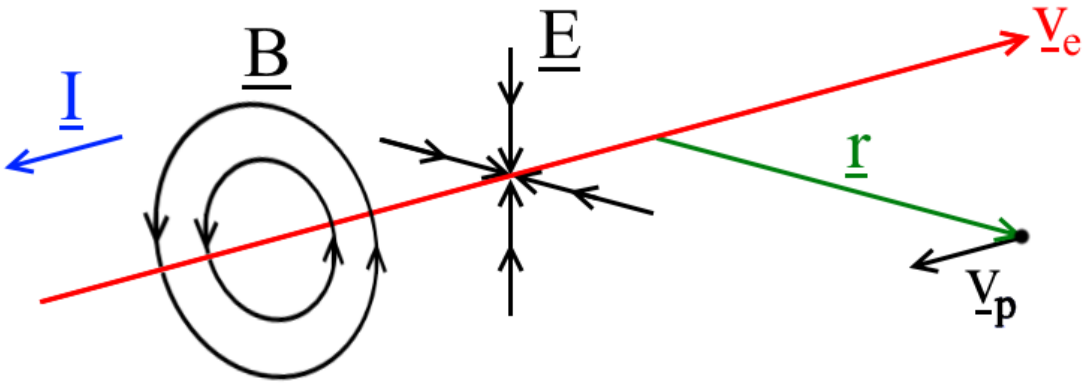

Figure 3.13: HEL modelled as an infinite line charge, indicating direction of the EM fields that influence proton motion. The magnetic field lines B, the electric field lines E, the HEL current I, the electron velocity v_e, the proton velocity v_p, and the radial displacement r are shown.

The electric field E and magnetic flux density B generated by an infinite line charge are well defined (see Appendix. B), and can be substituted into equation 2.3 to give the force on a proton with velocity v_p travelling parallel to the HEL at some transverse displacement r. Noting that the vectors B and v_p are perpendicular, we find the force on the proton in the lab frame to be:

$$F(r) = \frac{Iq(1 \pm \beta_e \beta_p)}{2\pi\epsilon_0 v_e \; r}. \tag{3.4}$$

Where I is the electron beam current, q is the proton charge, $\beta_e = \frac{v_e}{c}$ is the normalised electron velocity, and $\beta_p = \frac{v_p}{c}$ is the normalised proton velocity. The $\pm$ originates from the addition or subtraction of the magnetic flux density term from the electric field term in 2.3, which is dependent on the relative directions of proton and electron beams.

As the electric field E is generated by electrons, the direction of this force on a proton is attractive, i.e. towards the propagation axis of the charge, which in this case is the centre of the HEL beam.

We use the angular velocity to translate this force (see Appendix. B) into the maximum angular kick given to a proton interacting with the HEL:

$$\theta_{max}(r) = \frac{1}{4\pi\epsilon_0 c^2}\frac{2LI_r(1+\beta_e\beta_p)}{(B\rho)_p\beta_e\beta_p}\frac{1}{r}, \tag{3.5}$$

where I_r is the charge enclosed by the radius r, and L is the length of the HEL. We may define a function $f(r)$, which, for an ideal HEL with uniform charge density and radial symmetry, modulates the charge enclosed I_r in this expression to take into account the hollow cylindrical distribution of electrons:

$$f(r) = \begin{cases} 0, & r < R_{min} \\ \frac{r^2 - R_{min}^2}{R_{max}^2 - R_{min}^2}, & R_{min} < r < R_{max} \\ 1, & r > R_{max} \end{cases} \tag{3.6}$$

where:

r is the machine particle radius in transverse phase space,

R_{min} is the minimum HEL radius in transverse phase space, and

R_{max} is the maximum HEL radius in transverse phase space.

Thus the magnitude of the HEL kick on a proton is defined as a function of the protons transverse radius r, the beam rigidity of the machine beam $(B\rho)_p$, the HEL current I, length L, and the HEL and machine beam normalised velocities β_e and β_p respectively, in equation 3.7:

$$\theta_{kick}(r) = f(r) \cdot \frac{1}{4\pi\epsilon_0 c^2}\frac{2L_{HEL}I(1+\beta_e\beta_p)}{(B\rho)_p\beta_e\beta_p}\frac{1}{r}. \tag{3.7}$$

3.8.5 Non Round Beams

The HEL electron distribution is round in transverse space, this is due to the cylindrical cathode and is maintained by the solenoid field and space charge effects [67]. Therefore the HEL offers the greatest interaction with the beam halo at a position in the accelerator where the beam is round, that is to say where $\beta_x \approx \beta_y$. For positions where this is not the case, a large proportion of the halo may not interact with the HEL, thus diminishing its effectiveness as a collimation enhancer.

3.8.5.1 Elliptically Matched Operation

For operation of a HEL on a 'non-round' beam (*i.e.* where $\beta_x \neq \beta_y$) an 'elliptical' matching has been devised to attempt to maximise the overlap of the HEL with the

non-round beam halo.

Assuming dispersion to be negligible, we may illustrate the transverse footprint of the beam at a given position in the accelerator using an ellipse with semi-major and semi-minor axes (a, b respectively) being proportional to the square root of the beta functions in x and y. That is to say an ellipse with $a = n\sigma_x$ and $b = n\sigma_y$, where n is the number of sigma that denotes the minimum HEL radius. When $\beta_x \approx \beta_y$, this ellipse is a circle. For the non-round position in the HL-LHC, $\beta_x \approx 1.3 \cdot \beta_y$, and thus the beam footprint may be depicted as an ellipse with $a = 1.3 \cdot b$. As the HEL minimum radius R_{min} is set using σ_x, in reality this results in a radius that is 1.14 times too large in the y plane, as shown in Fig. 3.14. Thus the halo in one plane does not fully interact with the HEL, resulting in a diminished collimation enhancement.

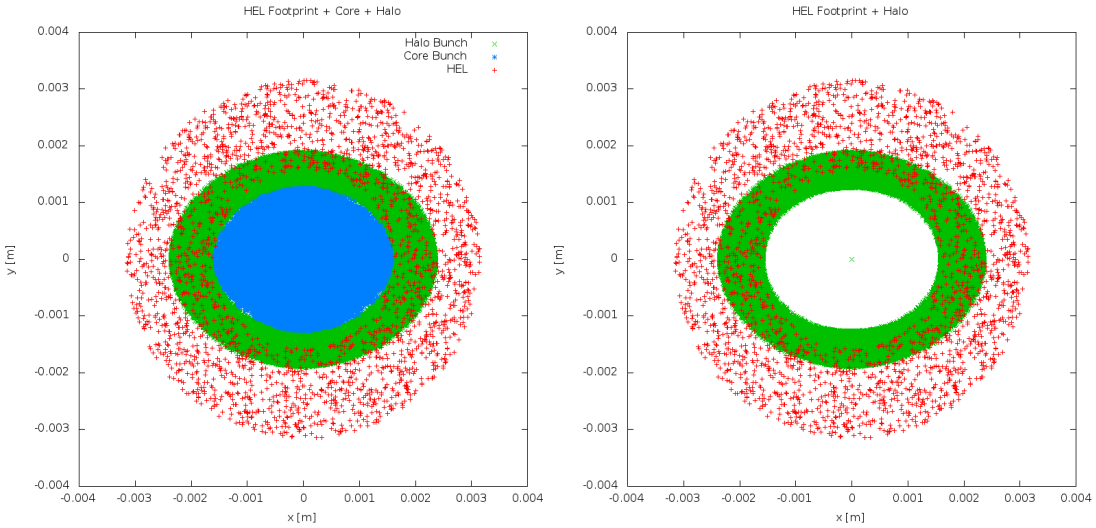

Figure 3.14: Diagram of the HEL intersection with a non-round beam footprint. The red points indicate the electrons in the HEL, the green points indicate the position of the halo (4-6 σ in x and y), and the blue points the core of the beam (0-4 σ in x and y). Here the HEL minimum and maximum radii are set to 4 and 8 σ_x respectively. The left plot shows the core, halo and HEL beams, the right plot omits the core for clarity.

By taking an extreme case we may derive an expression to modify the radii and offset of the HEL, in order for it to overlap with more of the halo in both planes. This is shown schematically in Fig. 3.15.

By setting R_{min} to meet the beam core ellipse at its semi-minor extremity (in this case the maximum y), and crossing both semi-major extremities (in this case the maximum x values), we may use simple trigonometry to find the magnitude of the required HEL inner radius, which we will label $R_{min\ (elliptical)}$:

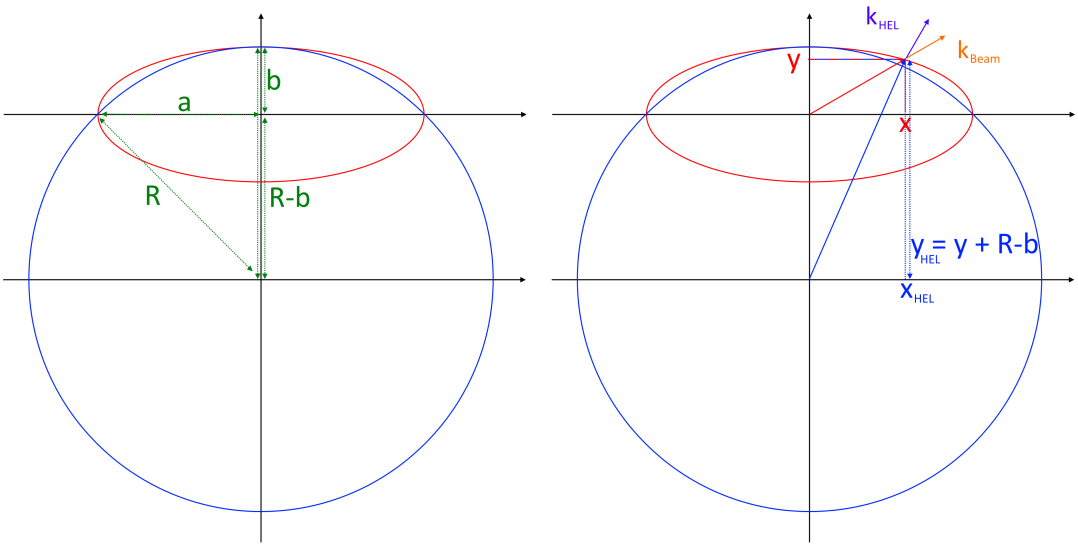

Figure 3.15: Diagram of an offset HEL minimum radius (blue), with a non-round beam core envelope (red) where the semi-major is much greater than the semi-minor axis. The left figure illustrates variables used to derive the shifted HEL centre and radii. The right figure shows the co-ordinates and kick in the beam frame and the HEL frame.

$$R_{min\ (elliptical)} = \frac{a^2 + b^2}{2b}. \tag{3.8}$$

We must also express the shift in co-ordinates (in this case in y), y_{shift}, as:

$$y_{shift} = y - R_{min\ (elliptical)} + b. \tag{3.9}$$

Thus we may use a HEL with minimum radius $R_{min\ (elliptical)}$, shifted to be centred at (x, y_{shift}), to maximise the overlap of the round HEL on a non-round accelerator beam where $\beta_x > \beta_y$. In order to set the HEL maximum radius R_{max} we use the fact that the ratio g:

$$g = \frac{R_{max}}{R_{min}}, \tag{3.10}$$

is a constant that depends only on the hardware (i.e. the cathode geometry), such that:

$$R_{max\ (elliptical)} = g \cdot R_{min\ (elliptical)}. \tag{3.11}$$

We also note from Fig. 3.15, that $R_{min\ (elliptical)}$ overlaps with the beam core. This

is undesirable, and is mitigated by using a scaling factor of $\sqrt{\frac{a}{b}}$, modifying equation 3.8:

$$R_{min\ (elliptical)} = \sqrt{\frac{a}{b}} \cdot \left(\frac{a^2 + b^2}{2b} \right). \tag{3.12}$$

The resulting 'matched' HEL is shown in Fig. 3.16.

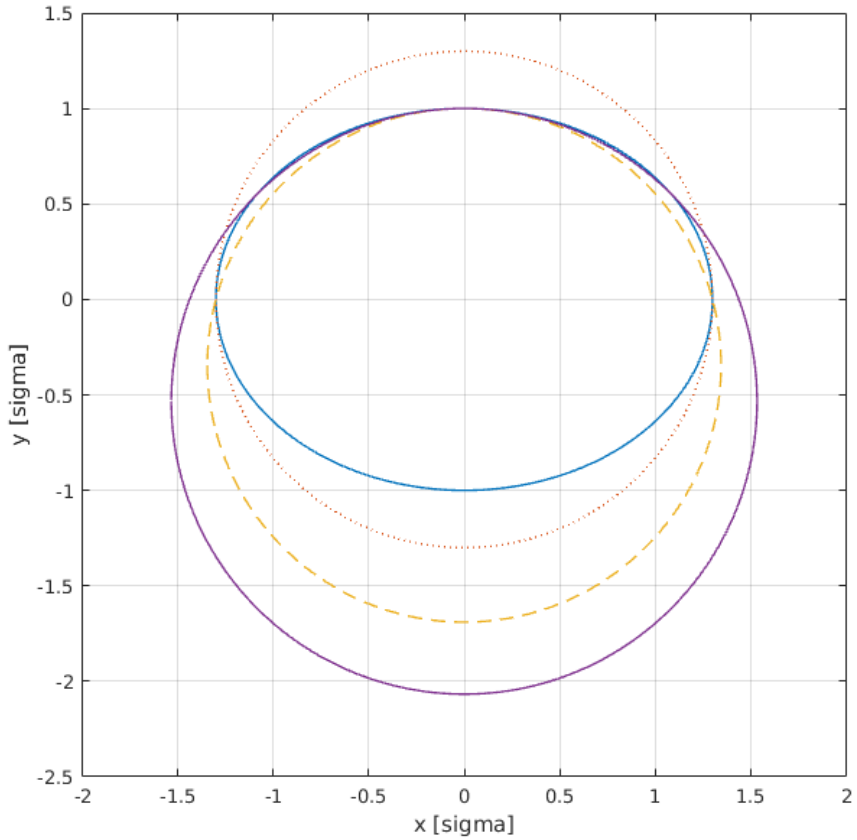

Figure 3.16: Beam core at non-round position (blue), original HEL R_{min} (red dotted), offset HEL $R_{min\ elliptical}$ (yellow dashed), and corrected and offset $R_{min\ elliptical}$ (purple), all shown in units of normalised σ where $\sigma_x = 1.3\sigma_y$, and $\sigma_y = \frac{1}{4}$.

In reality the beam footprint is cumulative over many turns, particles trace an ellipse in phase space and thus any space in the phase space ellipse may contain a particle at a given turn. This is simply an approach to improve the performance of the HEL as a collimation enhancer when forced to use it at a position in the accelerator where the beam is non-round. The effectiveness of this approach will be ascertained in Chapter 6.

3.8.5.2 Pogo Operation

The first attempt at dynamic operation in simulations, this mode translates the elliptically matched HEL beam such that it alternately touches the top and bottom transverse extremities of the beam core on a turn-by-turn basis, as shown in Fig. 3.17, in order to improve halo coverage.

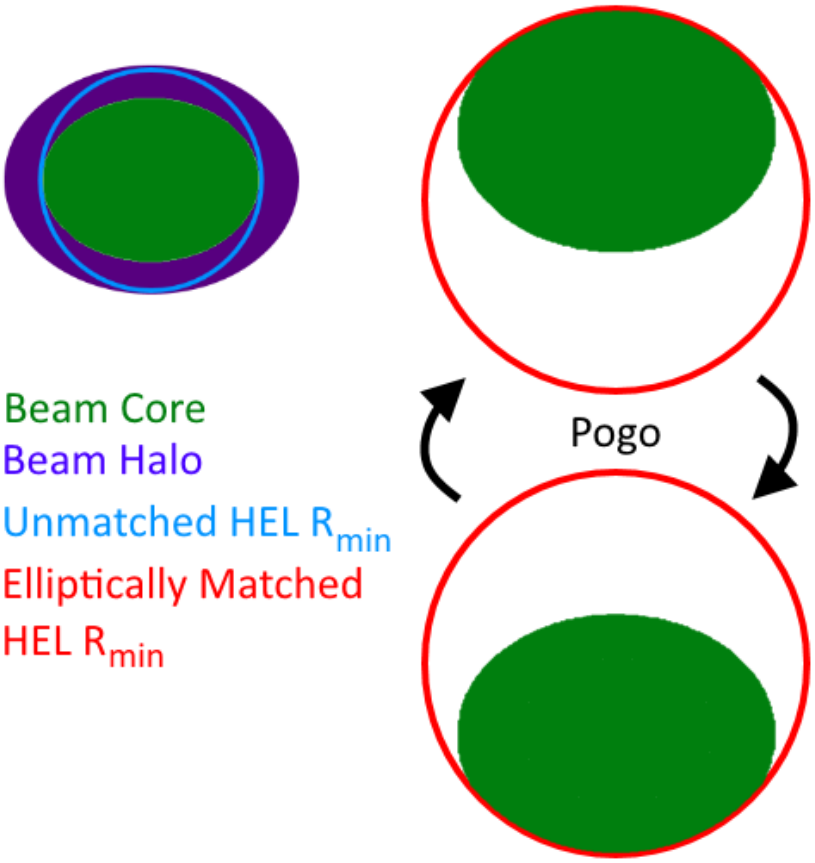

Figure 3.17: Dynamic Pogo operation of the HEL; the elliptically matched HEL is translated between the two transverse extremities in the semi-minor plane of the beam ellipse. The beam core is green, and the halo purple.

3.8.5.3 Hula Operation

This dynamic operation mode translates the elliptically matched HEL beam around the proton beam core. In order to reduced complexity a simple 4-step process was implemented in simulations, such that the HEL touches the top, right, bottom, then left transverse extremities of the beam core. This is shown in Fig. 3.18.

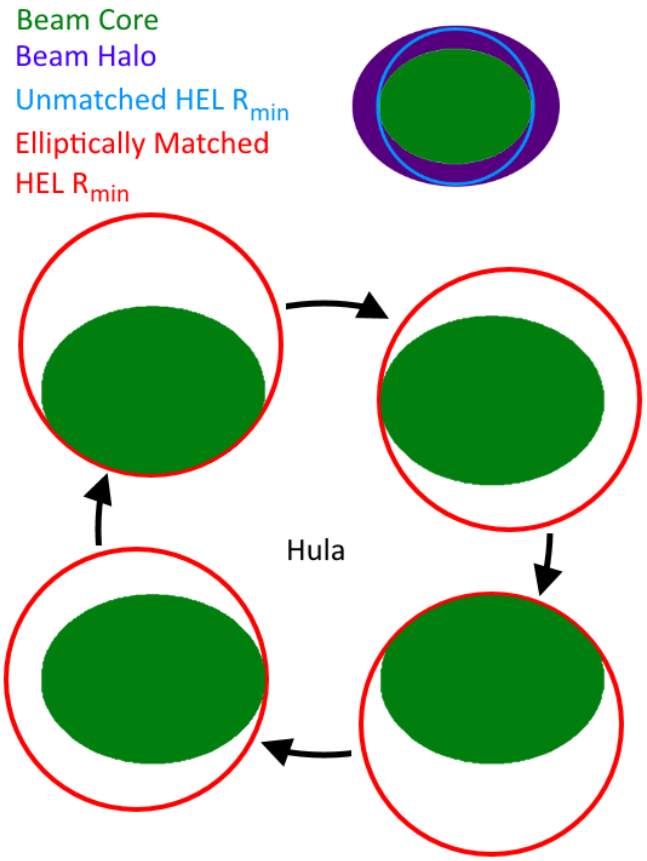

Figure 3.18: Dynamic Hula operation of the HEL; the elliptically matched HEL is translated between the four transverse extremities of the beam ellipse. The beam core is green, and the halo purple.

3.8.5.4 Close Hula Operation

Building on the Hula operation mode, this dynamic mode simply maintains a minimum HEL radius as it is translated, thus attempting to improve halo coverage as shown in Fig. 3.19.

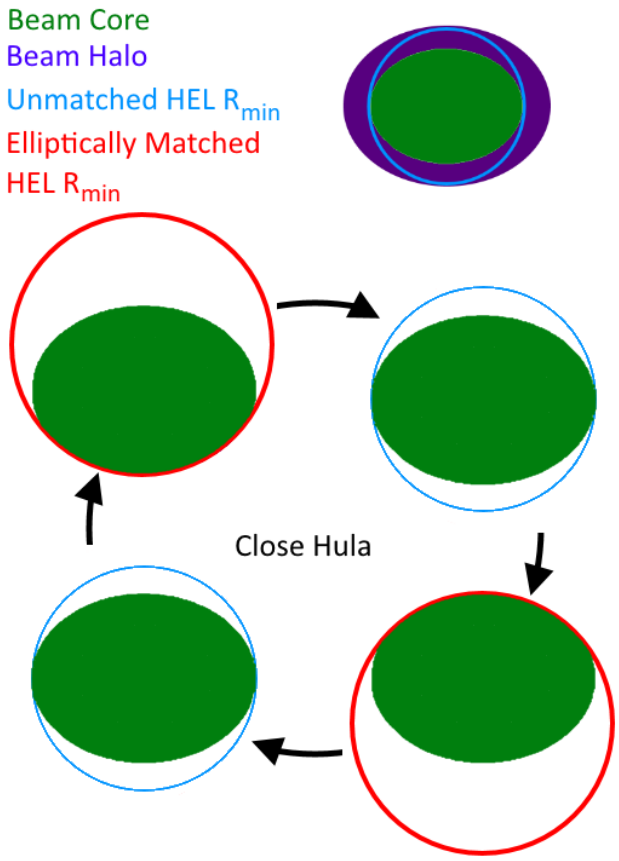

Figure 3.19: Dynamic Close Hula operation of the HEL; the HEL is translated to the four transverse extremities of the beam ellipse and has the smallest possible minimum radius, which is given by the elliptical matching at the semi-minor extremities, and the unmatched HEL radius at the semi-major extremities. The beam core is green, and the halo purple.

3.9 Modelling the Collimation System

The collimation system of the HL-LHC (and LHC) is vastly complicated, collimators are distributed over the entire synchrotron, offering multi-stage cleaning. The collimation system is designed to provide cleaning efficiencies of over 99.99%, thus loss locations must be identified with high statistical accuracy.

In order to predict limiting loss locations, and to understand measured losses, large scale simulations are required. In order to minimise run times, only the halo of the beam need be simulated. Other requirements include:

- accurate tracking of large orbit and energy deviations (of order of a few beam σ),
- many ($\gtrsim 10^5$) turn simulations,
- modelling of errors in jaw flatness and position, machine optics, and orbit.

The goal of a full simulation campaign is to determine the energy lost in the SC magnets for a given beam intensity impacting upon collimators. This is done using a number of tools, at CERN the first stage is done using SixTrack [18]. This includes particle tracking, collimation scattering, and loss locations for the production of loss maps. These maps identify the location and magnitude of proton losses after a beam halo has impacted upon the primary collimator. Typically a loss map simulation is done using $6.4 \cdot 10^6$ protons, for 200 turns. This is enough turns for the majority of particles to be lost, and provides the necessary statistics to resolve losses below the quench level [71]. An example measured qualification loss map is shown in Fig. 3.20, this is generated by forcing a partially populated beam to cross a machine resonance in order to generate 'artificial' losses (*i.e.* losses not occurring in normal operation), usually in a single plane - in this case horizontal betatron losses. This qualification verifies the efficiency of the collimation system, and is performed regularly during operation.

As collimators are the aperture restrictions of the machine we anticipate the highest losses to be in collimation insertions. As expected the highest losses in the LHC occur in the collimators in IR7, followed by losses in IR3.

After this simulation stage energy deposition studies are performed using FLUKA [19], and thermo-mechanical analysis is executed using codes such as ANSYS-AutoDyn [72].

In this thesis MERLIN takes the place of SixTrack, offering coupling to FLUKA in a similar manner to that of SixTrack (though it is not used), as well as updated proton-collimator scattering routines, novel and composite materials, and a more comprehensive HEL routine. Though these routines may be implemented in SixTrack, MERLIN provides ease of use; written in C++ (as opposed to multiple languages as

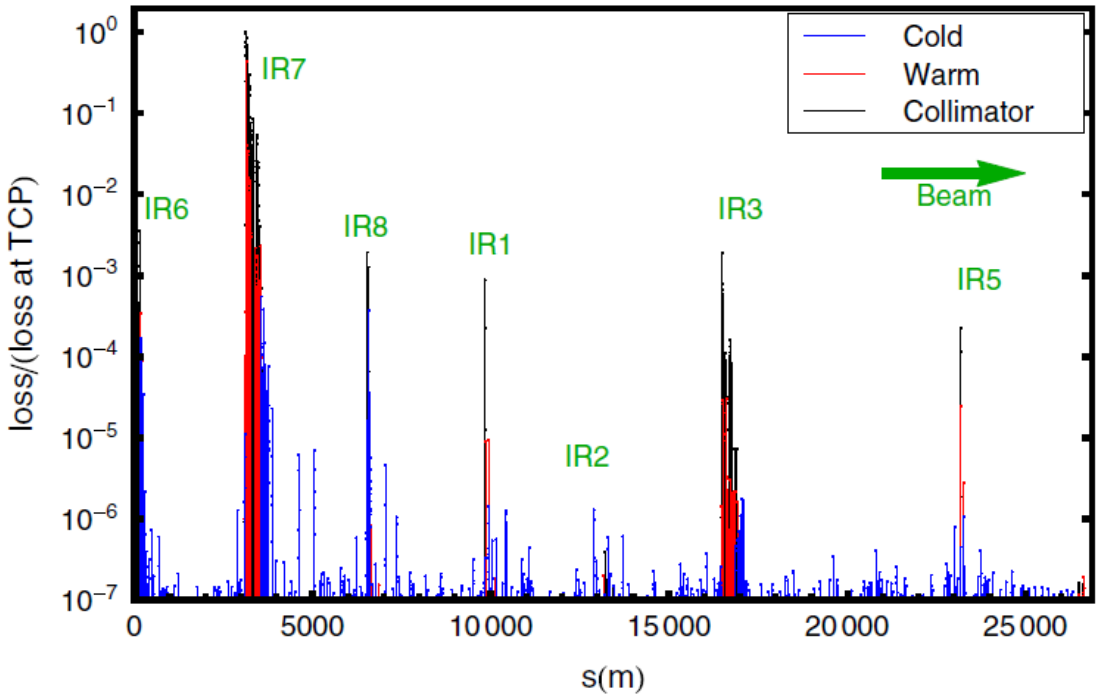

Figure 3.20: Example measured qualification loss map for the LHC at 3.5 TeV. This loss map is created by crossing a third order resonance to force particles onto the collimators. Losses are normalised to the highest loss (horizontal TCP in IR7). From [71]

in SixTrack) it is easier to read and supplement. MERLIN provides many features that SixTrack implements as post-processing (*e.g.* aperture checking). As well as this MERLIN is more adaptable because it is a library, for example the user may write their own physics, tracking, and input classes to override existing ones.

4

MERLIN 5.01

4.1 History & Overview

MERLIN is an accelerator physics library written in C++, created by Nick Walker at DESY in 2000, to study International Linear Collider [73] (ILC) beam delivery system ground motion [74]. Later the main linac and damping rings were added [75] necessitating wakefield, collimation, and synchrotron radiation processes [76]. As the ILC is an electron linac, the TRANSPORT maps were deemed acceptable for particle tracking. Later Andy Wolski added synchrotron functionality, including a module to calculate the Twiss parameters, closed orbit, and dispersion, and symplectic integrators for many turn simulations.

Many people were involved in the upgrades to MERLIN for LHC collimation, including Roger Barlow [77], Rob Appleby, Hywel Owen, Adina Toader [78], James Molson [79], Maurizio Serluca [80], Adriana Bungau, and Sam Tygier. The MERLIN collaboration have worked to meet a number of LHC collimation requirements, for previous work see [49].

4.2 Motivation

The current loss map tool for the LHC is SixTrack [18], a particle tracking code that was updated to include the K2 [81] scattering routines for collimation [82]. SixTrack is a thin lens tracker written in a number of Fortran versions, and is therefore difficult to modify. The scattering routines are a modification of the K2 code which is outdated. As SixTrack does not check if a particle has left the machine aperture, SixTrack collimation studies must perform this step using a post-processing tool, thus SixTrack is

not optimised for speed. The SixTrack hollow electron lens 'elens' routine was written by Valentina Previtali [68], but has not been used for a number of years, currently the routine is outdated, poorly documented and not well understood.

It was decided to update the MERLIN C++ library to include the requirements for a complementary simulation of the LHC collimation system. MERLIN is written in C++ making it modular, and easy to modify. It offers thick lens tracking, an on-line aperture check, and a number of physics processes. The scattering physics has recently been updated to include more advanced proton-nucleon elastic and single diffractive scattering [49]. These updates, together with the modifications detailed in this chapter, ensure that MERLIN offers a fast, accurate, and future-proofed tool for ultra-relativistic proton tracking, collimation, and a robust hollow electron lens process.

4.3 Modifications

In recent years MERLIN has been under the stewardship of Prof. Roger Barlow of the International Institute for Accelerator Applications (IIAA), University of Huddersfield. MERLIN was split into two branches, one developed at the University of Manchester and the other at Huddersfield. The two codes will be referred to as the 'Loss Map branch' [79], developed at Manchester by Robert Appleby, James Molson, Maurizio Serluca, and Sam Tygier, and the 'Hollow Electron Lens (HEL) branch' [83], developed at Huddersfield by Roger Barlow and Haroon Rafique.

In 2015 the two code branches were merged, combining the advanced scattering from the loss map code with the modular collimation and hollow electron lens processes of the HEL branch. Here we detail the result of merging both branches, MERLIN version 5.01 [84], which will henceforth be referred to simply as MERLIN. MERLIN is currently developed by Haroon Rafique, Sam Tygier, Alessandra Valloni, and James Molson, under the oversight of Roger Barlow and Rob Appleby.

MERLIN is prepared for operation with FLUKA in a similar way to SixTrack. It provides a more robust hollow electron lens routine, long term symplectic tracking, 6D tracking (including synchrotron motion), and the ability to read input files for the HL-LHC upgrade.

This chapter details the parts of MERLIN included and updated for collimation. Some of this was produced by others; the advanced single diffractive and elastic scattering routines (James Molson, Maurizio Serluca), the lattice functions, ring iterator, and symplectic tracker (Andy Wolski), as well as the existing framework of MERLIN. The majority of this work was produced by the author in collaboration with others,

either in modification of existing code or joint production, including but not limited to: the `CollimateParticleProcess`, the `MADInterface`, apertures and their construction, collimators and their construction, accelerator components, the `CollimatorSurvey`, materials, particle distributions and beam data, integrators, scattering processes, some physics content in particle scattering routines, and outputting non-collimator losses. Part of this work was produced solely by the author in agreement with the MERLIN collaboration: the proton collimation process, crab cavity failure process (not detailed here) and required phase advance functions, the FLUKA database, composite materials, the scattering model and all subsequent functions, the cross sections class, all collimation outputs with the exception of non-collimator losses, and the hollow electron lens process. Appendix D contains a more detailed code description with examples.

4.4 MERLIN 5.01

MERLIN version 5 is the result of merging the loss map and HEL codes. The definition of a MERLIN particle can be found in Chapter 2, equation 2.2, and the unit system used is defined in Table 4.1.

Quantity	**Unit**	**Symbol**	**Base Unit**
Energy	electron volt	eV	1 $[GeV]$
Distance	metre	m	1 $[m]$
Angle	radian	rad	1 $[rad]$
Time	second	s	1 $[s]$
Frequency	Hertz	Hz	1 $[Hz]$
Voltage	Volt	V	1 $[GV]$
Magnetic Flux Density	Tesla	T	1 $[T]$
Cross Section	barn	b	1 $[b]$

Table 4.1: MERLIN units defined in the `PhysicalUnits` namespace.

The following sections describe the physics used for a MERLIN collimation simulation. A MERLIN simulation consists of three main parts: creating an accelerator model, defining a beam, and selecting and using a tracker to transport the defined beam through the created accelerator. Physics processes may also defined and assigned to the tracker.

4.4.1 Accelerator Model

The model must define the type and location of accelerator elements. Each element contains an EM field, a geometry, and an aperture. Special cases exist, for example collimators also contain a jaw material which is required for scattering.

These accelerator components are stored as an `AcceleratorModel`, which is an ordered vector of `AcceleratorComponent`s, which is used by the `ParticleTracker` to set the integrators that describe the paths taken by individual particles as they travel through the elements.

The user may either define each individual element manually and construct a model using these, or standard MADX [85] thick lens TFS table may be passed through the `MADInterface` class to extract the `AcceleratorComponent`s. This is detailed in Appendix D.

4.4.2 Apertures

In order to correctly perform collimation, it is important that apertures exist for the entire accelerator. A MADX generated aperture list may be used to create the apertures for the accelerator. More information on apertures and their setup is given in Appendix D.

4.4.3 Lattice Functions

Apertures set using the MADX generated input file are set in units of metres, and centred on the perfect reference machine orbit (transverse (0,0)). For collimator jaw openings we use units of σ, which is proportional to the beam emittance and beta function on the given plane at the requested position in the lattice. For example in the x plane $\sigma_x = \sqrt{\beta_x \epsilon_x}$. This means that when setting collimator apertures we require the lattice functions.

MERLIN calculates these functions for all elements in the lattice, this is done by tracking a single particle and adjusting the initial conditions until the closed solution (*i.e.* the closed orbit) is found. Particles are then tracked with small deviations with respect to the closed orbit to construct the one turn map, and the other parameters are calculated using the particle co-ordinates at each element. The lattice functions are calculated from eigenvectors of the one turn map. A detailed description of the lattice function class is given in Appendix D.

4.4.4 Collimators

After constructing the `AcceleratorModel`, and assigning `Aperture`s to all non-collimator elements, we must give particular attention to the `Collimator`s in the lattice. As MERLIN has been developed to focus on the collimation system of the LHC, and now the HL-LHC, the `ApertureConfiguration` class is currently hard-coded not to set apertures for `Collimator` elements.

4.4.4.1 Collimator Database

The `ConfigureCollimators()` function is used to construct collimator apertures, for which a number of methods are available. The first is performed when giving this function only the `AcceleratorModel`, and uses the values given in the input file without taking into account the closed orbit or lattice functions of the machine. This method is therefore only valid when using units of metres in the input file for half gaps.

The more effective method is to use the overloaded `ConfigureCollimators()` that also takes the transverse emittances, and a pointer to the `LatticeFunctionTable`. Two flags exist that allow the user to take into account the closed orbit, and match collimator half gaps to the beta functions at the start and end of each collimator. By default these two flags are set to true, but may be switched off using the functions `MatchReferenceOrbit()` and `MatchBeamEnvelope()`.

The jaw half gap g, perpendicular half gap $g_{\perp}$, rotation angle α, and jaw tilts θ are illustrated in Fig. 4.1.

Using these definitions and the input file, the collimator apertures may be set in units of σ using equation 4.1:

$$\sigma = \sqrt{\beta_x \epsilon_x cos^2(\alpha) + \beta_y \epsilon_y sin^2(\alpha)}, \tag{4.1}$$

where β is the beta function in the given plane at the position of the collimator, ϵ is the beam emittance, and α is the jaw rotation in real transverse space. Note that we do not take into account the dispersion when setting the collimator apertures, instead the dispersion contribution is included in the jaw half gap value in σ [86]. In this manner the collimator jaws are set such that they follow a linear interpolation from the beam size ($\sqrt{\beta\epsilon}$) at the start, to the end of the collimator. This can be a difference of order 0.1 mm in the LHC depending on the position of the collimator in the lattice, and can be observed from the beta functions.

In SixTrack all apertures are thin *i.e.* have zero length, the values are input in units of σ, but define the aperture at the centre of the collimator. In order to compare with

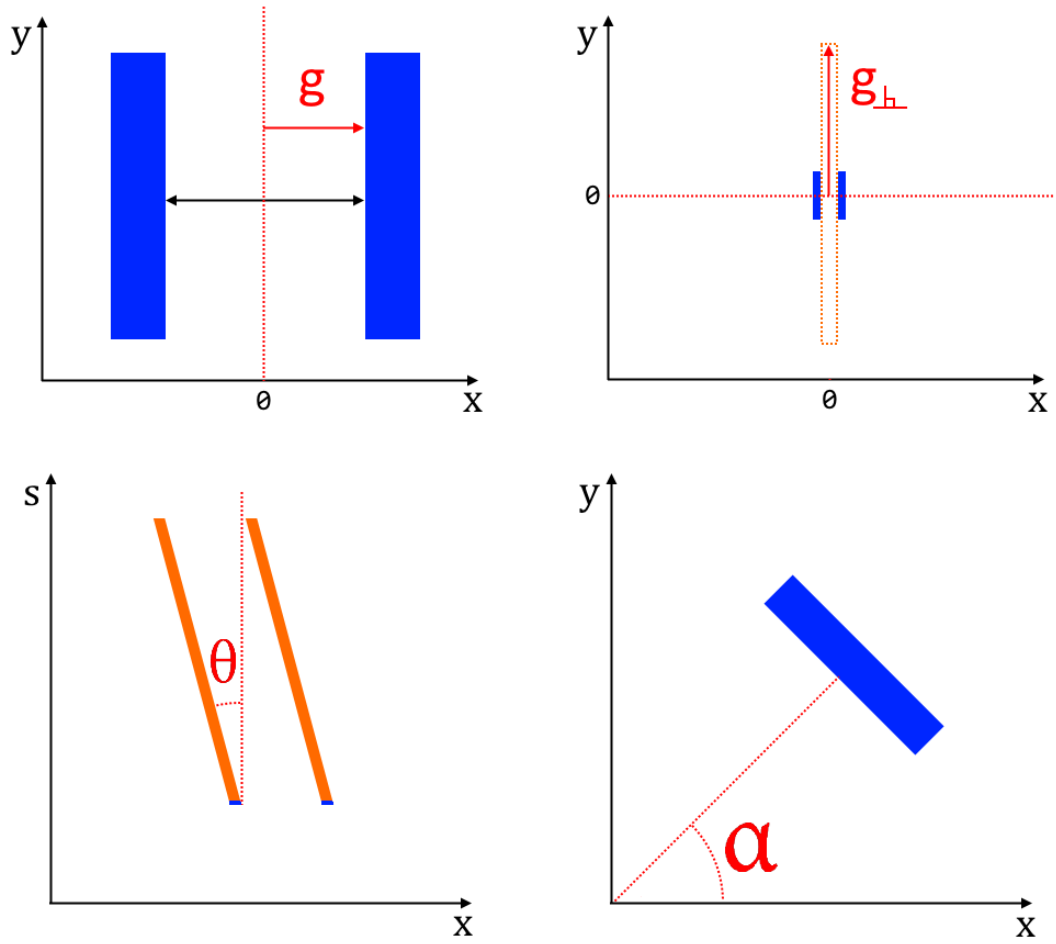

Figure 4.1: Definitions of jaw half gap g (top left), perpendicular half gap $g_\perp$ (top right), jaw tilt θ (bottom left), and jaw rotation α. Note that as the perpendicular gap is usually much larger than the jaw half gap, the effective aperture in real transverse space is often a slit, as depicted by the orange dotted rectangle in the top right figure. Blue shows the face of the jaws, orange the view from above.

SixTrack the function `UseMidJawHalfGap()` was added to the `CollimatorDatabase`. This function sets a flag which calculates the beta function at the centre of the collimator and uses it to set a collimator aperture that is constant throughout the length of the collimator.

The different methods of setting the collimator jaw half gaps are shown in Fig. 4.2. This is plotting using the `CollimatorSurvey` output, which is detailed in Appendix D.

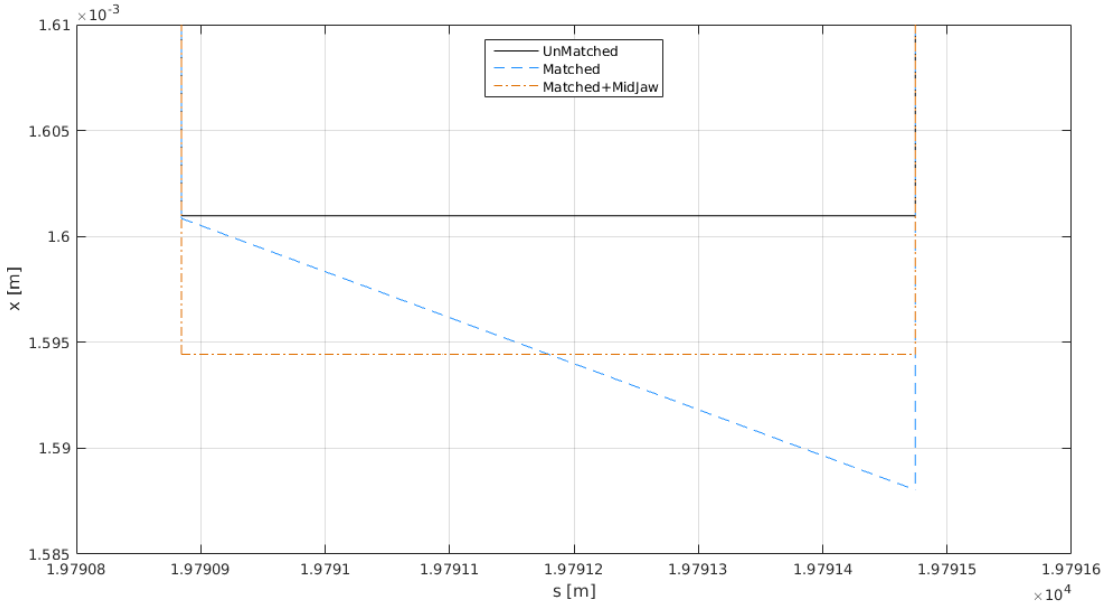

Figure 4.2: `CollimatorSurvey` output used to show the three different methods for setting collimator jaw half gaps. The unmatched method in black uses the β function at the start of the collimator, the matched method in blue adapts the opening to the beam envelope to maintain a constant half-gap in units of beam σ, and the MidJaw option in orange imitates SixTrack by using the beta function in the centre of the collimator.

4.4.4.2 Materials

The `Material` class stores all material properties required for scattering. A material may be created using the overloaded constructor, or by creating an empty `Material` and adding attributes individually. The attributes and expected units stored in `Material` are shown in Table 4.2. More information on this class is given in Appendix D.

The `MaterialDatabase` class provides a dictionary of standard materials (listed in Table 4.3) that may be used without the user defining the corresponding `Material` object. This dictionary is detailed in Appendix D.

Pure elements are stored as `Material` objects, but other types of materials may be stored as `CompositeMaterials`, or `Materials`. The difference between these is covered in the following section and the difference in implementation is explained in the collimation section.

Attribute	Symbol	Expected Unit	MERLIN name
Name	-	-	Name
Symbol	-	-	Symbol
Atomic Number	Z	-	AtomicNumber
Atomic Mass	A	amu	AtomicMass
Elastic Nucleus Reference Cross Section	σ_E	b	SixtrackElasticNucleusCrossSection
Rutherford Nucleus Reference Cross Section	σ_R	b	SixtrackRutherfordNucleusCrossSection
Inelastic Nucleus Reference Cross Section	σ_I	b	SixtrackInelasticNucleusCrossSection
Total Nucleus Reference Cross Section	σ_{tot}	b	SixtrackTotalNucleusCrossSection
Density	ρ	$\frac{kg}{m^3}$	Density
Conductivity	σ	$\frac{S}{m}$	Conductivity
Radiation Length	χ_0	$\frac{g}{m^2}$	RadiationLength
Mean Excitation Energy	$\bar{I}$	eV	MeanExcitationEnergy
Electron Density	n_e	m^{-3}	ElectronDensity
Nuclear Slope	b_n	$\frac{GeV}{c^2}$	SixtrackNuclearSlope

Table 4.2: Material attributes, symbol, and the expected unit for `Material` construction in MERLIN.

Material	Symbol
Carbon (graphite)	C
Beryllium	Be
Copper	Z
Aluminium	Al
Lead	Pb
Boron	B
Oxygen	O
Tungsten	W
Carbon Diamond	CD
Copper Carbon Diamond	CuCD
CFC AC150K	AC150K
Inermet 180	IT180
Glidcop	GCOP
Molybdenum Carbide	Mo2C
Molybdenum Carbide Graphite	MoGr

Table 4.3: Materials currently available in the `MaterialDatabase` dictionary, and the symbols used as accessors.

4.4.4.3 Composite Materials

In order to model novel composite materials for collimator jaw upgrades, a new `CompositeMaterial` class has been created in MERLIN. This allows the user to create a composite material from existing or user defined `Materials`. These constituent `Materials` may be pure elements or other composites. The class itself is based on the `MaterialMixture` class from the loss map code, however `MaterialMixture` does not provide the required functionality, and as a result is not compatible with the new `CollimateProtonProcess`.

`CompositeMaterial` contains a number of accessing and setting functions as in `Material`, constituents may be added by mass (m_i) or number (n_i) fractions using the `AddMaterialByMassFraction()` or `AddMaterialByNumberFraction()` functions.

A full treatment of the calculation of composite material properties can be found in Appendix C. The mass fraction m_i of the composite may be calculated from the number fraction n_i as shown in equation 4.2:

$$m_i = \frac{n_i A_i}{\sum_i n_i A_i}, \tag{4.2}$$

where A_i is the atomic mass of the constituent i. Conversely the number fraction n_i may be calculated from the mass fraction m_i as shown in equation 4.3:

$$n_i = \frac{\left(\frac{m_i}{A_i}\right)}{\sum_i \left(\frac{m_i}{A_i}\right)}. \tag{4.3}$$

After all constituents have been added, the `Assemble()` function calculates composite material properties. The following properties always return a weighted average of the constituent material properties: atomic mass, atomic number, electron density, plasma energy, mean excitation energy, stopping power $\left(\frac{dE}{dx}\right)$, and radiation length. We may define the mean atomic number $\bar{Z}$ [45],

$$\bar{Z} = \sum_i n_i \cdot Z_i, \tag{4.4}$$

where n_i is the number fraction of constituent i, which has atomic number Z_i. The mean atomic mass $\bar{A}$ is defined as a similarly weighted property [45],

$$\bar{A} = \sum_i n_i \cdot A_i, \tag{4.5}$$

where A_i is the atomic mass of constituent i.

Using these definitions we can define the electron density n_e, mean excitation energy $\bar{I}$, and the radiation length χ_0 for the composite material, as shown in Appendix C.

When handled by the new `CollimateProtonProcess`, the user may select one of two ways to handle all `CompositeMaterial`s.

The first is to treat them as homogeneous materials with imaginary nuclei that represent a weighted average of all constituents, this is the approach taken for 'new materials' in SixTrack [45], and was created for comparison with SixTrack. A comparison of composite materials created in this manner and their properties in MERLIN and SixTrack is given in Table 4.4. The nuclear reference cross sections for a composite are calculated using equation 4.6:

$$\sigma_{pN} = \sum_i n_i \sigma_{pN\ i}, \tag{4.6}$$

where $\sigma_{pN\ i}$ is the reference cross section for the corresponding nuclear process in the constituent i. A comparison of the cross sections for composites in MERLIN and SixTrack can be found in Table 4.5.

This approximation is required for the calculation of the total mean free path λ_{tot} for composite materials.

The second and more appropriate method of treating `CompositeMaterial`s is to use the homogeneous approach for bulk scattering, *i.e.* MCS and ionisation, but select a weighted random constituent for point like processes. By default MERLIN uses this approach, the function `ScatteringModel::SetComposites(bool)` can be used to switch to the first approach for SixTrack comparison by using the argument '`false`' or '0'.

Composite properties can be checked using the `VerifyMaterial()` function which checks all constituent materials as well as the composite, and returns a `false` boolean if there is an unset or unreasonable value.

For comparison the reference cross section values calculated in MERLIN were overwritten with those used in SixTrack. The resulting values are shown in Table 4.4 and Table 4.5 as **MERLIN modified**. The differences between the two codes are mostly negligible, where this is not the case they are justified below or in Appendix C.

We note that the proton nucleon cross sections for elastic and single diffractive scattering are not dependent on the reference nuclear cross sections, and so are not affected by the differences between MERLIN and SixTrack.

When using MERLIN scattering with the proper treatment of composite materials,

Property	MoGr	CuCD	Glidcop	Inermet180
Z				
MERLIN	6.611	11.896	28.824	67.66
SixTrack	6.653	11.898	28.823	67.657
A				
MERLIN	13.44	25.23	63.145	166.7
SixTrack	13.532	25.238	63.149	166.68
$\rho\ [\frac{g}{cm^3}]$				
MERLIN	2.5	5.4	8.93	18
SixTrack	2.5	5.4	8.93	18
$\sigma_{el}\ [\frac{MS}{m}]$				
MERLIN	1	12.6	53.8	8.6
SixTrack	1	12.6	53.8	8.6
$\chi_0\ [m]$				
MERLIN	0.1214	0.03164	0.01443	0.00385
SixTrack	0.11931	0.03162	0.01442	0.00385
$\frac{dE}{dx}$				
MERLIN	0.7188	1.981	2.685	5.576
$b_n\ [\frac{GeV}{c^2}]$				
MERLIN	79.69	121.3	223.59	427
SixTrack	76.665	114.961	208.669	392.137
$\lambda_{tot}\ [m]$				
MERLIN	0.2301	0.1259	0.0892	0.0577
SixTrack	0.2484	0.1356	0.0942	0.0603
MERLIN Modified	0.2257	0.1259	0.0892	0.0577

Table 4.4: Properties of composite materials in MERLIN and SixTrack [45].

the only effect of the composite properties is seen in the multiple Coulomb scattering and ionisation calculations (`ScatteringModel::EnergyLoss()` and `ScatteringModel::Straggle()`). These functions rely on the electron density, plasma density, mean excitation energy, and radiation length, which are all treated with standard methods for composites. The mean free path is also used as a step length in the collimation process function `CollimateProtonProcess::DoScatter()`.

Once we have constructed our lattice in the form of an `AcceleratorModel`, set the machine aperture using the `ApertureInterface`, and configured the material and settings for all collimators via the `CollimatorDatabase`, we have a complete accelerator. The next step is to create a beam of particles to simulate, and a tracker to track these particles and assign physics processes to.

Property	MoGr	CuCD	Glidcop	Inermet180
$\sigma_{pN\ tot\ ref}$				
MERLIN	0.3545	0.572	1.246	2.546
SixTrack	0.362	0.572	1.246	2.548
$\sigma_{pN\ I\ ref}$				
MERLIN	0.244	0.370	0.765	1.525
SixTrack	0.247	0.370	0.765	1.473
$\sigma_{pN\ R\ ref}$				
MERLIN	$11.9{\cdot}10^{-5}$	0.000449	0.00151	0.006807
SixTrack	$9.4{\cdot}10^{-5}$	0.000279	0.001385	0.005737
$\sigma_{pN\ tot}$				
MERLIN	0.3827	0.6068	1.293	2.611
MERLIN modified	0.3902	0.6068	1.293	2.613
$\sigma_{pN\ E}$				
MERLIN	0.07459	0.1584	0.4217	0.939
MERLIN modified	0.0796	0.1583	0.4217	0.993
$\sigma_{pn\ E}$				
MERLIN	0.04487	0.05536	0.07516	0.1039
MERLIN modified	0.04487	0.05536	0.07516	0.1039
$\sigma_{pn\ SD}$				
MERLIN	0.001879	0.02318	0.03148	0.0435
MERLIN modified	0.001879	0.02318	0.03148	0.0435

Table 4.5: Cross sections of composite materials in MERLIN and SixTrack citequarantaprivate, all values are given in barns.

Accessor	Component	Index
x	x	0
xp	x'	1
y	y	2
yp	y'	3
ct	ct	4
dp	δ	5
type	Type of last particle scatter	6
s	Location in lattice	7
id	Individual particle ID	8
sd	Single diffractive flag	9

Table 4.6: Components of the `PSVector` class.

4.5 Bunch Definition

MERLIN stores particles as `PSVector`s, a class that contains the particle co-ordinate vector as components, as well as a number of other variables, all of which are detailed in Table 4.6.

The `ParticleBunchConstructor` class is used to create an initial bunch matched to the machine lattice functions at any chosen injection position. In order to do this a `Beam Data` object must be created and fed to the bunch constructor. The user must specify a `ProtonBunch` in construction or the default `ElectronBunch` will be constructed.

4.5.1 Distributions

MERLIN provides a number of bunch distributions, all of which are stored in the `ParticleBunchConstructor`. The majority are described in [49], this will not be repeated here. The first particle in a bunch is always the reference particle. The user may specify the construction of the bunch via an input file as shown in Fig. D.19, the file must contain six columns, listing the co-ordinates: x, x', y, y', ct, δ in that order, with a new particle on each line, and the first particle as the reference particle.

The bunch constructor matches the beam to the lattice functions as shown in equation 4.7, where the individual transformations are given in equations 4.8 - 4.10.

$$\mathbf{p}_f = \mathbf{DBA}\ \mathbf{p}_i, \tag{4.7}$$

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ D_x & D_{x'} & D_y & D_{y'} & 0 & 1 \end{pmatrix} \tag{4.8}$$

$$\mathbf{B} = \begin{pmatrix} \sqrt{\beta_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{\beta_x}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{\beta_y} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{\beta_y}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.9}$$

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -\alpha_x & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\alpha_y & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.10}$$

For the study of the HEL in the LHC and HL-LHC, a HEL halo distribution was created. This is a simple halo bunch that is populated between $\sigma_{x_{min}}$ and $\sigma_{x_{max}}$ in x, and $\sigma_{y_{min}}$ and $\sigma_{y_{max}}$ in y, thus a matched halo distribution in xy phase space is generated, as shown in Fig. 4.3. The minima and maxima are specified using the `BeamData` class as discussed previously. All other coordinates are matched to regular beam parameters and the optics of the machine at the point of injection in the simulation.

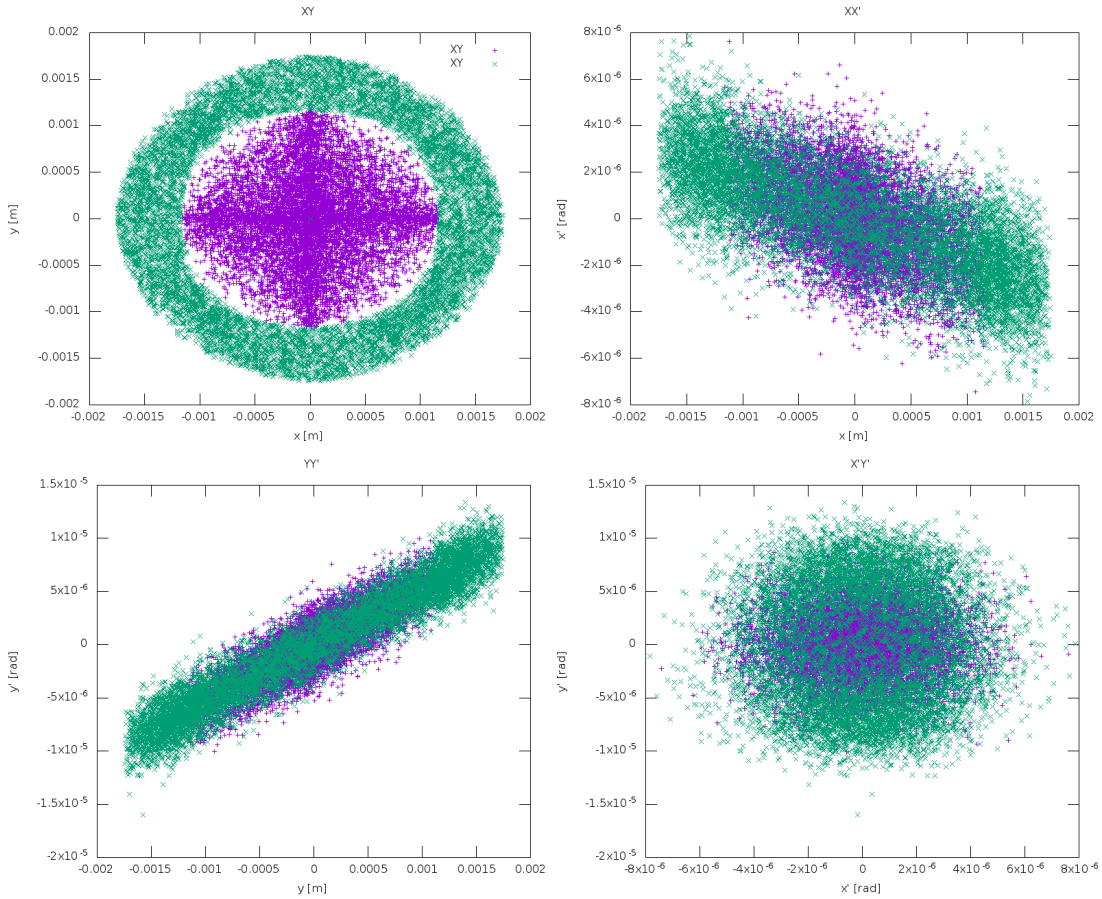

Figure 4.3: HEL halo distribution in xy, xx', yy', and $x'y'$ phase space. Purple points are a 'core' bunch populated between 0-4 σ_x and σ_y, green points are a 'halo' bunch populated between 4-6 σ_x and σ_y. This bunch is created at an 'injection' position of HEL in the nominal LHC.

4.6 Tracker

Integrators define the tracking for a particle through a single element of the accelerator lattice, and can be stored using any method that is compatible with the defined particle bunch co-ordinates. Integrators are described in Appendix D.

The `TRANSPORT` integrators are described in Chapter 2, and detailed in Appendix D.

4.6.1 `SYMPLECTIC`

The `SYMPLECTIC` integrator set was added to MERLIN by A. Wolski, however as it was not used it did not behave as expected with MERLIN 5.01. It was found that all particles were lost unexpectedly when running LHC loss map simulations, the tracks of these particles are compared with those from the `TRANSPORT` integrator in Fig. 4.4.

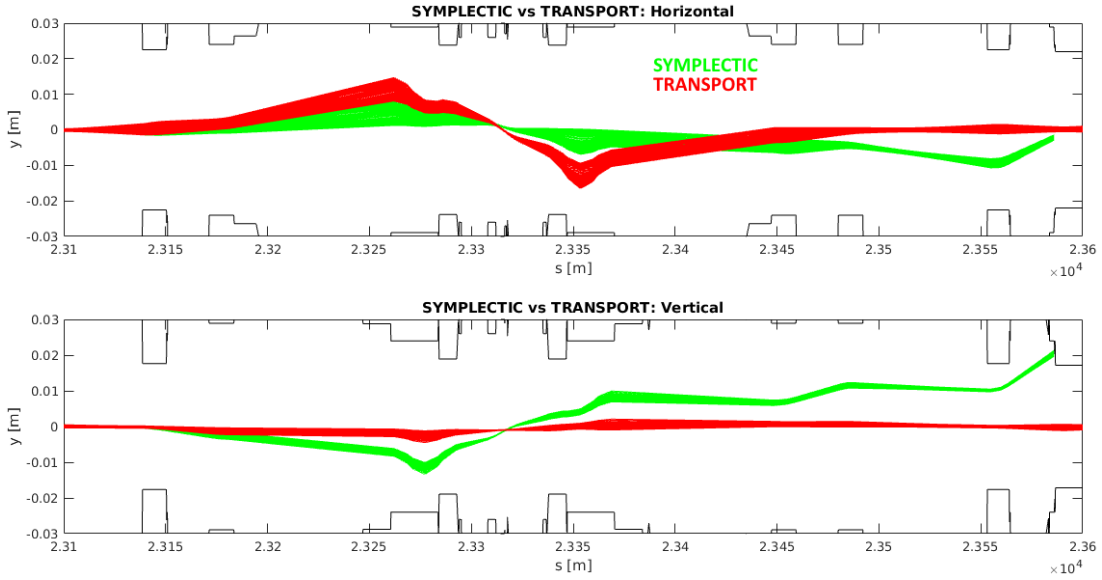

Figure 4.4: Particle tracks in the 6.5 TeV LHC indicating the incorrect path taken by protons when using the original `SYMPLECTIC` (green) tracker, compared to the correct path when using the `TRANSPORT` (red) tracker. Particles being tracked by the `SYMPLECTIC` tracker are unexpectedly lost in a vertical aperture of the machine.

To ascertain the cause of the incorrect tracking, each integrator in the set was debugged. It was evident that a number of functions required updating, in particular the map for multipoles was not being applied properly. Once resolved, particle tracking with the `SYMPLECTIC` tracker became almost identical to that with the `TRANSPORT` tracker as expected.

4.6.2 Tracking Output

The particle tracker has the capability to take an output class as detailed in Appendix D. This may be used to output individual particle tracks, which can be used to observe the paths taken by particles that have scattered differently in the collimator jaws. This tool was also used to identify bugs in the `SYMPLECTIC` tracker.

This type of output is particularly useful for observing the effect of different scattering types on a proton impacting upon a collimator. Figure 4.5 shows the tracks of a proton halo that impacts the primary horizontal collimator in IR7 in the 6.5 TeV LHC lattice. Recording the first scattering type shows us how the dispersion suppressor losses are caused by single diffractive scattering in the collimator jaws. The same discovery in SixTrack led to the explanation of the unexpected losses in LHC operation, the consequence of which are the dedicated TCLD dispersion suppressor collimators for the HL-LHC upgrade.

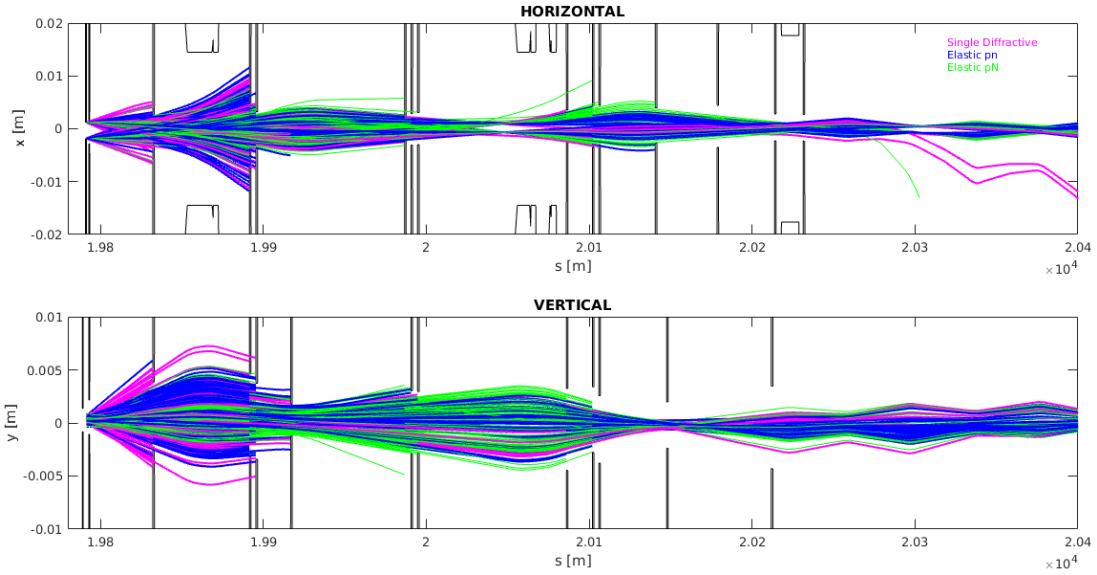

Figure 4.5: Tracks of a proton halo distribution that impacts upon the horizontal primary collimator. Particles that undergo an initial scattering type of proton-nucleon single diffractive (magenta), proton-nucleon elastic (blue), and proton-nuclear elastic (green) are shown in the horizontal and vertical planes in IR7 of the 6.5 TeV LHC. Black lines indicate the apertures of collimators and other elements. Where coloured lines end they are absorbed by an aperture or undergo inelastic scattering in a collimator.

4.6.3 Synchrotron Motion

Synchrotron motion provides another mechanism for particle loss, as particles outside the RF bucket will likely be lost in an aperture. The off-momentum collimation insertion in IR3 of the LHC is designed for this purpose. The RF bucket, and synchrotron motion in MERLIN is demonstrated in Fig. 4.6, which is from an LHC simulation with collimation disabled.

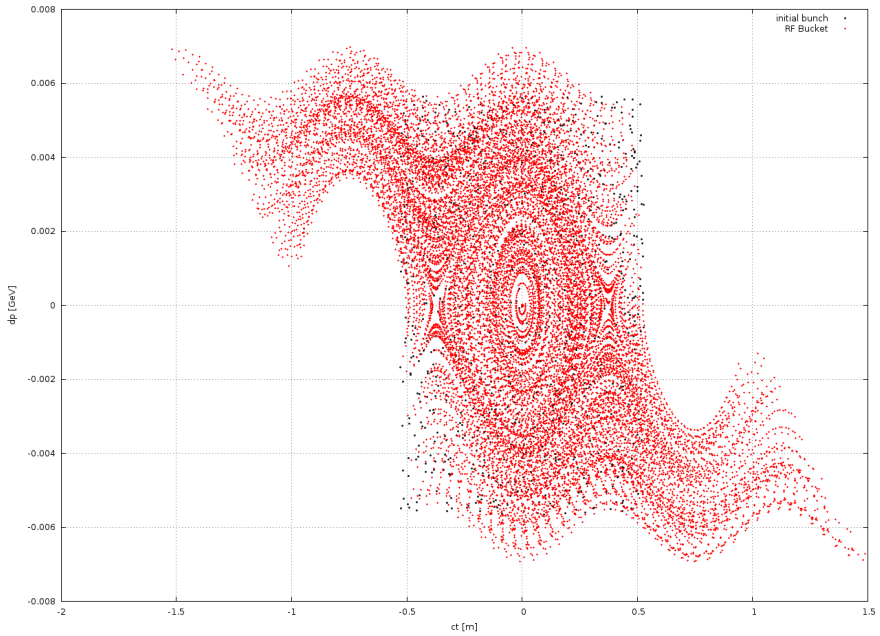

Figure 4.6: Poincaré section in ct, δ phase space of a large initial distribution (black) over 100 turns in the LHC (red), showing RF bucket and synchrotron motion.

We may compare Fig. 4.6 with Fig. 4.7, which is from a simulation with collimation enabled. We observe that particles with large momentum deviation are lost in the apertures of the machine, and note that the initial distribution in both figures is not physical, and used only for a demonstration of these effects.

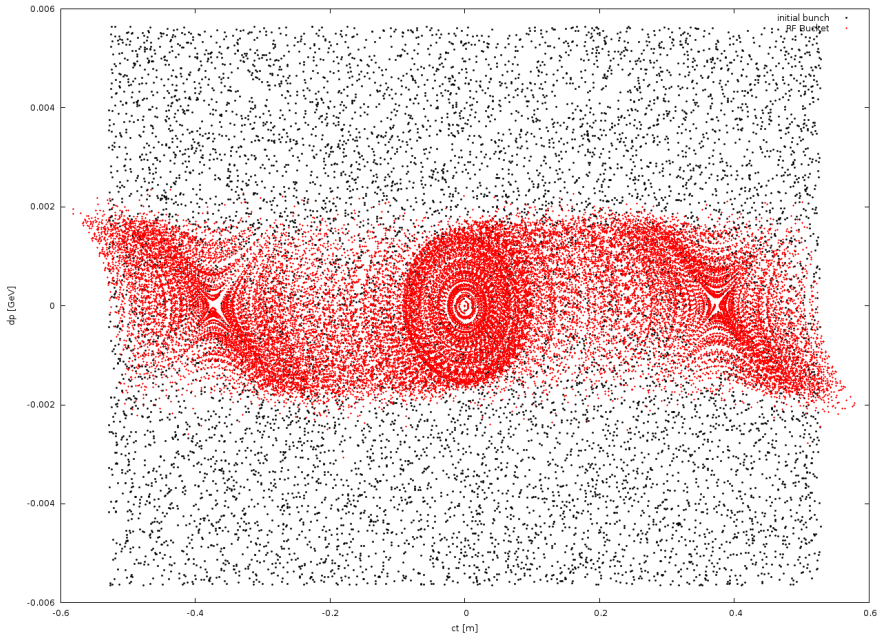

Figure 4.7: Poincaré section in ct, δ phase space of a large initial distribution (black) over 100 turns in the LHC (red), showing the effect of collimation on particles with large energy deviation.

4.7 Physics Processes

Physics processes may represent any mechanism, the process may be run at selected accelerator elements, or all. Collimation and the action of the hollow electron lens as a collimation enhancer are performed in MERLIN using dedicated processes.

4.8 Collimation Process

The collimation process has been developed by the MERLIN collaboration for the past few years with the goal of LHC loss map production. In the loss map code the collimator scattering was performed in the `ProtonBunch` via `CollimateParticle Process`. In the HEL code an independent collimation process was developed using a child class of `CollimateParticleProcess` named `CollimateProtonProcess`. This uses an optimised modular system of classes that are detailed in this section, and used as default in MERLIN 5.01. The code flow for proton collimation in MERLIN 5.01 is shown in Fig. 4.8.

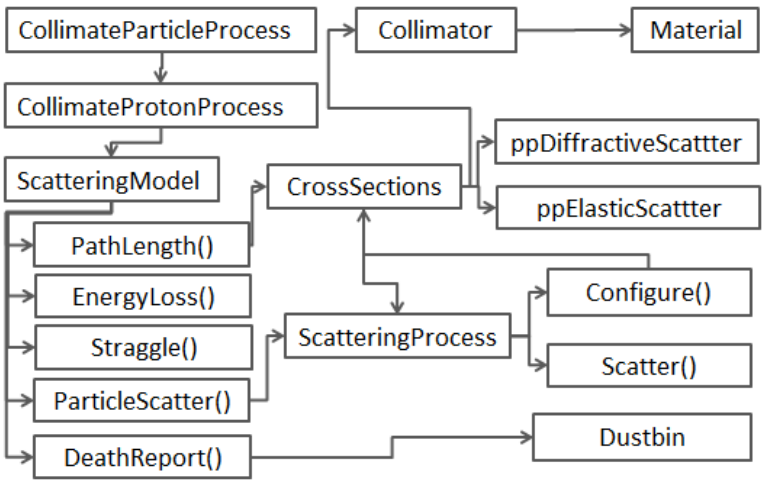

Figure 4.8: The code flow of proton collimation in MERLIN 5.01. `CollimateParticle Process` is attached to the tracker, if the particles being tracked are protons, the `Collimate ProtonProcess DoScatter` function is called when a particle is outside the machine aperture. This in turn calls the `ScatteringModel` functions that perform calculation of the path length, the energy loss and particle kick due to MCS in the collimator material over this length, and any point-like scattering. In order to do this the `ScatteringModel` functions have access to the collimator material, the cross sections for scattering in that material, the `Scattering)Process` functions that perform point-like scattering, and various output classes.

A complete example of creating the proton collimation process is shown in Appendix D.

4.8.1 `CollimateParticleProcess`

`CollimateParticleProcess` is called at every element in the tracker (if the process has been attached to the tracker). This parent collimation class was originally developed for ILC collimation which is relatively simple as the ILC is an electron machine. By default this process is for electrons, a remnant from the original application. The process uses the standard functions for a physics process.

`SetCurrentComponent()` records the first element name and s position, which are used as an increment marker for the turn (which is used in the `CollimateProton Process` for a number of outputs), the internal turn integer is incremented each time the process is called at the start element. `SetCurrentComponent()` also checks that the bunch contains particles, and that the current element has an aperture, if so then the process will run. If the element has no aperture at all, no collimation is performed. Collimation with scattering only occurs for `Collimator` elements with a `Collimator Aperture`, for all other elements the aperture is the surface of a black absorber which will cause the loss of any particle that crosses the aperture boundary.

`CollimateParticleProcess` contains a step size which may be set with the `Set OutputBinSize()` function, `DoProcess()` iterates through these steps and calls the `DoCollimation()` function at each iteration.

`DoCollimation()` contains the main collimation routine, it begins by running a pre-check, iterating through the bunch and checking if any particles are outside the aperture using the aperture `PointInside()` function. This is done because the process copies the entire bunch into one that is operated upon, only surviving particles are copied back into the original bunch at the end of the process. This is currently the best way of removing lost particles, though memory intrusive it is preferable to removing particles from a list. Performing the pre-check minimises the time taken for collimation as for most elements no particles are lost. If no particles are outside the aperture process is exited, otherwise a marker is set and the pre-check ends, full collimation is then performed for the marked particle and all subsequent particles.

The process then iterates through the bunch and uses the `Aperture::PointInside()` function to check if each particle is outside the aperture. If the element is not a collimator and doesn't have a collimator aperture, the particle is lost, and its co-ordinates are stored for output. The `ScatterAtCollimator()` function may be used to turn off scat-

tering in collimators in order to use them as black absorbers similar to non-collimator elements.

In `DoCollimation()` the `DoScatter()` function is called for a particle outside the aperture of a collimator. This function is overloaded, in the HEL code if using electrons, the simple electron scattering routine is called. In the loss map code this function calls the corresponding function in `ProtonBunch`. In MERLIN 5.01 the electron scattering routine is no longer used. In the HEL code, and MERLIN 5.01, the user declares a `CollimateProtonProcess` rather than `CollimateParticleProcess`, and the `CollimateProtonProcess::DoScatter()` function performs proton collimation. In a similar manner the user may define collimation processes for any given particle to be tracked in MERLIN, as a child class of `CollimateParticleProcess`.

If the particle is lost due to scattering in the collimator jaw, `DoScatter()` returns true, and the particle is stored for output. If the particle survives, it is added to a new particle bunch, which becomes the particle bunch at the end of the process, this is less expensive computationally than removing individual particles from a bunch. For non-collimator elements the lost particles are tracked through the element to find the lost position to the nearest step size bin.

4.8.2 `CollimateProtonProcess`

When using protons `CollimateProtonProcess` must be attached to the `Particle Tracker`. This class uses the functions from `CollimateParticleProcess` except where they are redefined, as it is a child class. `DoScatter()` is the only one to be redefined in `CollimateProtonProcess`. The `ScatteringModel` is set here with the `SetScattering Model()` function.

`DoScatter()` controls proton scattering in the collimator jaw. From here, the `ScatteringModel` functions that perform multiple Coulomb scattering (MCS), ionisation energy loss, and point like scattering are called. `DoScatter()` starts by checking which outputs have been selected in the `ScatteringModel`, whether they have been selected for the current collimator, and preparing accordingly. Next the function loops over the bin size, or the remaining collimator length as appropriate, iterating over steps which are calculated using a random variant of the mean free path in the material using the `ScatteringModel::PathLength()` function.

The proton is tracked along the step size, MCS and ionisation energy loss is performed for the path length, and a check is performed to ascertain whether or not the MCS has caused the proton to return to the aperture, or if the proton travels to the

end of the bin or collimator without interacting with a nucleus in the collimator jaw. Point like nuclear scattering is then performed using the `ScatteringModel::Particle Scatter()` function. If the proton undergoes inelastic scattering, or its energy is below a certain threshold (for example 1 GeV), the particle is lost.

Before detailing the `ScatteringModel` class and its functions, we must explain the `CrossSections` class which calculates and stores cross sections for each material, and the `ScatteringProcess` class, which provides the mechanism for nuclear and nucleon scattering.

4.8.3 `CrossSections`

For optimisation the `CrossSections` class calculates and stores all cross sections for a given material. The components of the class are shown in Table 4.7. These cross sections are called by the `ScatteringProcess` classes when performing point like scattering, and in `ScatteringModel::PathLength()` to retrieve the total mean free path λ_{tot}. By using this class to compute and save the cross sections, MERLIN minimises computation time at the cost of an inexpensive amount of memory.

`CrossSections` stores the advanced `ppElasticScatter` and `ppDiffractiveScatter` classes (for more details on these classes see [49]), allowing access to them during the collimation processes.

MERLIN provides 5 predefined combinations of `ScatteringProcesses`, one of which may be selected using the `ScatteringModel::SetScatterType()` function, which takes an integer that corresponds to the columns in Table 4.8, detailed in the following section. For each combination the cross sections and mean free path must be calculated in the appropriate manner. As well as an advanced treatment of single diffractive and nucleon elastic scattering, MERLIN contains SixTrack+K2 like scattering (henceforth referred to as ST-like) for comparison and validation purposes [49]. This is not up-to-date with the scattering used in the most recent collimation version of SixTrack, however it provides useful functionality, and demonstrates the ease with which a user may port their own methods into MERLIN.

The probability that a proton of momentum P will interact with a nucleus, or nucleon of that nucleus, is given by the nuclear (σ_{pN}) and nucleon (σ_{pn}) cross sections respectively. Cross sections are measured in units of area, the standard unit used is the barn b, where 1 $b = 10^{-28} m^2$.

In order to calculate nucleon cross sections the number of free nucleons n_n is used to approximate the number of nucleons in a nucleus that a proton may scatter off of,

Component	Symbol	Parameter
E0	E_0	Reference Energy
sig_pN_tot_ref	$\sigma_{pN\ tot\ ref}$	Reference Total Nuclear Cross Section
sig_pN_inel_ref	$\sigma_{pN\ I\ ref}$	Reference Inelastic Nuclear Cross Section
sig_pN_R_ref	$\sigma_{pN\ R\ ref}$	Reference Rutherford Nuclear Cross Section
sig_pN_tot	$\sigma_{pN\ tot}$	Total Nuclear Cross Section
sig_pN_inel	$\sigma_{pN\ I}$	Inelastic Nuclear Cross Section
sig_pN_el	$\sigma_{pN\ E}$	Elastic Nuclear Cross Section
sig_R	σ_R	Rutherford Nuclear Cross Section
sig_pp_tot	$\sigma_{pp\ tot}$	Total Proton-Proton Cross Section
sig_pp_el	$\sigma_{pp\ el}$	Elastic Proton-Proton Cross Section
sig_pp_sd	$\sigma_{pp\ SD}$	Single Diffractive Proton-Proton Cross Section
sig_pn_el	$\sigma_{pn\ el}$	Elastic Proton-Nucleon Cross Section
sig_pn_sd	$\sigma_{pn\ SD}$	Single Diffractive Proton-Nucleon Cross Section
lambda_tot	λ_{tot}	Total Mean Free Path
elastic_diff	$\Delta\sigma_{pp\ el}$	Difference in σ with and without Coulomb Peak [49]
density	ρ	Material Density
atomic_mass	A	Atomic Mass
atomic_number	Z	Atomic Number
scat_type	-	Scattering Type
symbol	-	Material Symbol

Table 4.7: Components of the `CrossSections` class.

it is defined as:

$$n_n = 1.618\ A^{\frac{1}{3}}, \tag{4.11}$$

where A is the atomic mass of said nucleus.

When using MERLIN scattering, the total proton-proton cross section $\sigma_{pp\ tot}$, *i.e.* the probability of a proton interacting with a proton in the collimator jaw, is given by a parameterisation provided by the particle data group (PDG) [87]. In the case of the ST-like (SixTrack like) scattering, equation 4.12, which is derived from a fit of experimental data from the PDG, is used [88]:

$$\sigma_{pp\ tot\ ST} = \sigma_{pp\ tot\ ref} \cdot \left(\frac{P_0}{P_{ref}}\right)^{0.05788}, \tag{4.12}$$

where $\sigma_{pp\ tot\ ref} = 40\ mb$ is the reference cross section, at a reference energy $P_{ref} = 450\ GeV$, and P_0 is the beam momentum: 7 TeV in the case of the nominal LHC. In all cases proton-neutron scattering is considered equal to proton-proton scattering, which has been measured more extensively, giving more experimental data.

Two main types of proton-nucleon scattering are considered. The first is an elastic scatter, in which the beam proton transfers some of its momentum to the nucleon, is scattered at some angle θ, and both nucleons survive. The second is a single diffractive scatter, in which one of the protons is excited to a higher mass state, which will dissociate and be lost. For the purpose of collimation we only consider the case of the target nucleon dissociating, as it may result in a large momentum transfer from the incident proton, and a large angle scatter. This is particularly interesting as it is thought to be the main cause of unexpected proton losses in dispersion suppressor regions after collimation and interaction insertions, as part of the betatron halo becomes an off-momentum halo [53].

MERLIN generates the differential cross sections for elastic and single diffractive nucleon scattering, and integrates them to provide more accurate cross sections than those used in SixTrack+K2, this is done using a fit to selected relevant data (see [49]). In the ST-like case the elastic proton-proton cross section $\sigma_{pp\ el}$ is derived from a fit of experimental data provided by the PDG [88]:

$$\sigma_{pp\ el\ ST} = \sigma_{pp\ el\ ref} \cdot \left(\frac{P_0}{P_{ref}}\right)^{0.0479}, \tag{4.13}$$

where $\sigma_{pp\ el\ ref} = 7\ mb$ is the reference cross section at the reference energy $P_{ref} = 450\ GeV$. For single diffractive scattering in the ST-like case, the method proposed by

Goulianos [89] is simplified to give the cross section:

$$\sigma_{pp\ sd\ ST} = A\ ln(0.15\ E^2_{com}), \tag{4.14}$$

where E^2_{com} is the squared centre of mass energy, and $A = 0.00068$ is a constant.

In both scattering methods, the proton-nucleon cross sections are given by multiplying the number of free nucleons n_n with the respective proton-proton cross section:

$$\sigma_{pn\ sd} = n_n \sigma_{pp\ sd}, \tag{4.15}$$

$$\sigma_{pn\ el} = n_n \sigma_{pp\ el}. \tag{4.16}$$

The total nuclear cross section for both MERLIN and ST-like scattering is given by summing the total proton-nucleon cross section with the reference total nuclear cross section:

$$\sigma_{pN\ tot} = \sigma_{pN\ tot\ ref} \cdot n_n(\sigma_{pp\ tot} - \sigma_{pp\ tot\ ref}), \tag{4.17}$$

where $\sigma_{pN\ tot\ ref}$ is defined for the material by the user or computed in the `materials` class, the reference proton-proton cross section $\sigma_{pp\ tot\ ref} = 0.04\ b$, and $\sigma_{pn\ tot}$ is the computed proton-nucleon cross section that corresponds to the MERLIN or ST-like method.

Nuclear elastic scattering is similar to the nucleon case, however the interaction is with the nucleus rather than a nucleon. The elastic nuclear cross section is the remainder once all other nuclear cross sections are computed and subtracted from the total cross section:

$$\sigma_{pN\ el} = \sigma_{pN\ tot} - \sigma_{pN\ I} - \sigma_{pn\ el} - \sigma_{pn\ sd}, \tag{4.18}$$

this is the same for both MERLIN and ST-like scattering.

Electromagnetic scattering from the nucleus is treated as Rutherford scattering. Both MERLIN and ST-like scattering use the user input (or calculated) Rutherford cross sections. The final cross section required is the inelastic, which defines the probability that the scattered proton does not survive the point-like scattering process. In the collimation process we treat an inelastically scattered proton as lost, though in reality a number of secondary particles may shower due to this interaction, this is outside the scope of MERLIN. MERLIN uses the reference nuclear inelastic cross section, however

the ST-like method scales this as shown in equation 4.19:

$$\sigma_{pN\ I\ ST} = \sigma_{pN\ I\ ref} \left(\frac{\sigma_{pN\ tot}}{\sigma_{pN\ tot\ ref}} \right). \tag{4.19}$$

Each individual cross section is used by, and associated with its own `Scattering Process`, the `CrossSections::ConfigureCrossSections()` function is called in the constructor, and performs calculation of the cross sections using the appropriate methods. The `GetTotalMeanFreePath()` function returns the mean free path computed using the corresponding cross sections (*i.e.* MERLIN or ST-like). MERLIN calculates the mean free path using equation 4.20:

$$\lambda_{tot} = \frac{A}{(\sigma_{pN\ tot} + A \cdot \Delta_{el}\sigma_{pp\ el}) \cdot \rho N_a}, \tag{4.20}$$

where Δ_{el} is the difference in the proton-proton elastic cross section when including or excluding the low t Coulomb peak from the fit to data [49]. For ST-like scattering, equation 4.21 is used:

$$\lambda_{tot} = \frac{A}{\sigma_{pN\ tot} \cdot \rho N_a}, \tag{4.21}$$

Thus the probabilities for each `ScatteringProcess` are calculated once per material, and stored inside a `CrossSections` object, that may be accessed where required in the collimation process.

4.8.4 `ScatteringProcess`

`ScatteringProcess` is a virtual class for individual point-like scattering processes. It contains a pointer to the `Material` and `CrossSections` classes, the process cross section, the beam energy, and two functions: `Configure()` and `Scatter()`.

We use the Lorentz invariant Mandelstam variables to perform particle scattering, the momentum transfer t, and the centre of mass energy squared E^2_{com}, where:

$$t = -2P^2(1 - cos\theta) \approx -(P\theta)^2, \tag{4.22}$$

where P is the beam momentum, and θ is the polar angle:

$$\theta = tan^{-1}\sqrt{x'^2 + y'^2}. \tag{4.23}$$

We simplify the momentum transfer by assuming an elastic interaction, and using the small angle approximation. The second Mandelstam variable is defined as:

$$s = E_{com}^2 = 2m_pP + 2m_p^2 \approx 2m_pP, \tag{4.24}$$

where the first expression is used for MERLIN scattering, and the second (a simplification) is used for ST-like scattering. The centre of mass energy squared is the total available energy for the interaction, and is often referred to as s.

The `Configure()` function takes the `Material` and `CrossSections` pointers as arguments, gets the cross section and beam energy from the `CrossSections` object, and calculates the momentum transfer t. The `Scatter()` function takes a `PSvector` object (*i.e.* a particle) and performs scattering using the overloaded `ScatterStuff()` functions. These functions take a number of variables, the simplest form uses the momentum transfer t to calculate the polar angle θ, and applies this angular scatter to the particle, the second form adds the nuclear recoil, and the third is used when the energy change is calculated in the `Scatter()` function.

MERLIN contains a number of `ScatteringProcesses`, including the ST-like variants, those currently available are shown in Fig. 4.9.

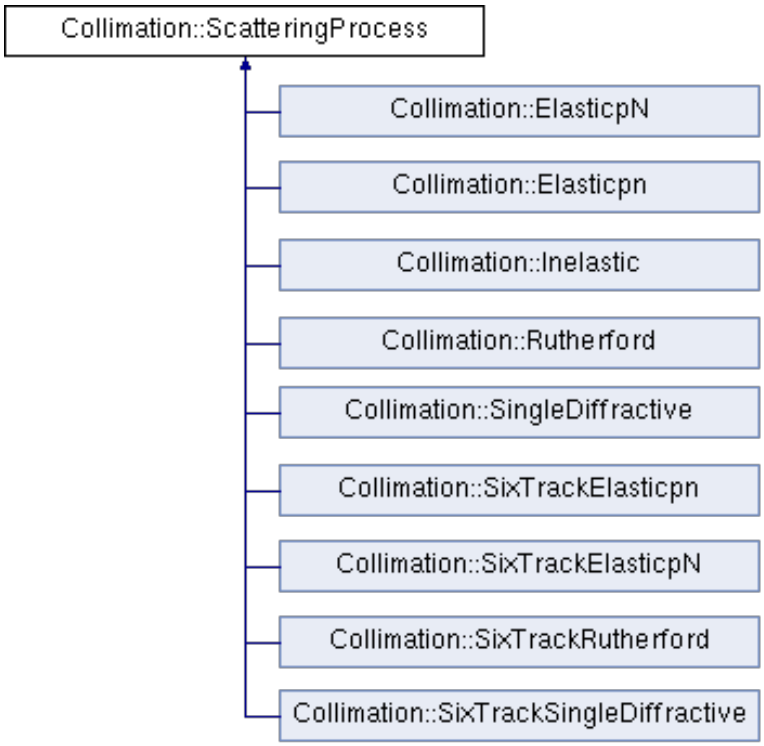

Figure 4.9: Scattering processes currently available in MERLIN 5.01.

The user may select a predefined combination of these `ScatteringProcess`es using the `ScatteringModel::SetScatterType()` function, which takes an integer corresponding to the combinations displayed in Table 4.8.

Process	**0**	**1**	**2**	**3**	**4**
Rutherford	ST	ST	ST	ST	M
pn Elastic	ST	M	ST	ST	M
pN Elastic	ST	M	ST	ST	M
Single Diffractive	ST	ST	M	ST	M
Ionisation	ST	ST	ST	M	M

Table 4.8: Preset combinations of `ScatteringProcess`es and ionisation in MERLIN 5.01. ST refers to the ST-like process, M to the MERLIN process, and all combinations include an inelastic process.

Each particle, as a `PSVector`, has a `type` member, which corresponds to the last type of point-like scattering that the particle has undergone. At construction of the initial bunch all particles are set to -1 (no scatter), this changes every time the particle passes through the `ScatteringProcess::Scatter()` function, or the functions for ionisation or MCS. Table 4.9 shows the `type` integers and the corresponding scattering processes. We note that a particle is only recorded as undergoing MCS or ionisation if point-like scattering has not occurred, this exception is allowed because particles always experience MCS and ionisation along any path length travelled in a material.

Scatter Type	**Label**
None	-1
Ionisation	0
Inelastic	1
pN Elastic	2
pn Elastic	3
pn Single Diffractive	4
Multiple Coulomb Scattering	5
Rutherford	6

Table 4.9: Scattering types and integer labels given to the particle `type`.

The individual `ScatteringProcess`es currently available in MERLIN are detailed in the following subsections.

4.8.4.1 Rutherford

SixTrack and MERLIN treat Rutherford scattering in the same way, however MERLIN includes a nuclear recoil effect. The momentum transfer is given by equation 4.25:

$$t_{Ruth} = \frac{t_{min}}{1 - R_u}, \tag{4.25}$$

where R_u is a uniformly random generated number between 0 and 1, and $t_{min} = 0.9982 \cdot 10^{-3}\ GeV^2$ [3].

4.8.4.2 pn Elastic

For proton-nucleon elastic scattering, MERLIN generates the momentum transfer t in the advanced `ElasticScatter` class. For ST-like scattering, the nuclear slope b_{pp} is used to take into account the energy dependence of the differential cross section:

$$b_{pp} = 8.5 + 1.086\ ln(E_{com}^2), \tag{4.26}$$

This slope is part of the approximation of the differential cross section for this process, of the form $\frac{d\sigma}{dt} = \sigma\ b(s)\ e^{-b(s)\ |t|}$, where b represents the slope of the exponential, σ is the total cross section, and s is the centre of mass energy squared. This approximation and its merits and assumptions are discussed in [49], however it is considered appropriate for the energy range of the LHC (450 GeV to 7 TeV). From this the momentum transfer t is given:

$$t_{pn\ el} = -\frac{ln(R_u)}{b_{pp}}. \tag{4.27}$$

Once again MERLIN includes a recoil effect whereas the ST-like method does not.

4.8.4.3 pN Elastic

Proton-nuclear elastic scattering is similar for MERLIN and ST-like scattering, with the usual nuclear recoil included in MERLIN scattering. In both cases the nuclear slope b_N is used to approximate the energy dependence of the cross section:

$$b_N = b_{N\ ref} \cdot \frac{\sigma_{pN\ tot}}{\sigma_{pN\ tot\ ref}}. \tag{4.28}$$

The momentum transfer for this process is then computed using equation 4.29:

$$t_{pN\ el} = -\frac{ln(R_u)}{b_N}, \tag{4.29}$$

where R_u is a uniform random number between 0 and 1.

4.8.4.4 Single Diffractive

For MERLIN scattering, the momentum transfer t and recoil mass m_{rec} are generated in the advanced `DiffractiveScatter` class. In the ST-like case, the recoil mass (that of the excited mass state) is given by:

$$m_{rec} = e^{\left(R_u\ ln\sqrt{E^2_{com}}\right)}. \tag{4.30}$$

It modifies the momentum transfer using a simple fit of the data, which gives a different slope parameter b depending on the mass range [88]. This is performed using the piecewise equation 4.31:

$$b = \begin{cases} 2b_{pp}, & M^2 < 2\ GeV^2 \\ \frac{1}{36}(106 - 17M^2)b_{pp}, & 2 \leq M^2 \leq 5\ GeV^2 \\ \frac{7}{12}b_{pp}, & M^2 > 5\ GeV^2 \end{cases} \tag{4.31}$$

The momentum transfer is then computed using:

$$t_{pn\ sd} = -\frac{ln(R_u)}{b}, \tag{4.32}$$

Both methods include recoil from the excited mass state, and use an energy loss δ_{sd} of:

$$\delta_{sd} = \frac{m^2_{rec}E}{E^2_{com}}, \tag{4.33}$$

where E is the incident proton energy.

4.8.4.5 Inelastic

For most materials the total cross section is dominated by inelastic interactions. As mentioned previously these can result in a shower of secondary particles, but for the purposes of collimation in the LHC (*i.e.* loss map production), in order to minimise run

time, and to prevent the recreation of a full shower code, MERLIN treats an inelastic interaction as a proton loss.

4.8.5 `ScatteringModel`

The `ScatteringModel` class contains the functions required for performing collimation. A predefined or user created combination of `ScatteringProcesses` may be used, and are handled by the `ScatteringModel` in order to compute cross sections, path lengths, and perform bulk (ionisation and MCS) and point-like scattering. The `Scattering Process` must be attached to the `CollimateProtonProcess` as shown in Fig. D.25.

4.8.5.1 `PathLength`

Protons travel a finite distance through a material before colliding with a material nucleus. Along this distance the proton loses energy as it collides with electrons, ionising atoms in its path, scattering in angle and position via MCS. The distance travelled is referred to as the path length in MERLIN, and it is calculated in the `PathLength()` function.

In order to calculate the path length, the mean free path must first be computed. `ScatteringModel` initialises the treatment of materials by creating a single `Cross Sections` object for each `material`. These are stored in a map, in this way all cross sections and the mean free path calculated in the `CrossSections` object need only be computed once per `material`. For a `CompositeMaterial`, there are two options available to the user. The first is the default, where a `CrossSections` object is created for each constituent element in the composite, and each is stored in (if not already added to) the `CrossSections` map. As well as this a `CrossSections` class is created for the composite as a homogeneous mixture of its constituents, including cross sections for all processes, and the mean free path for the composite, this is the one used for the path length calculation. The second option is the ST-like method of only creating a single `CrossSections` object for the composite, treating it as a homogeneous mixture for all cross sections as well as the mean free path. This option may be enabled by the user with the `ScatteringModel::SetComposites(bool)` function with a `false` argument.

The path length is not constant, the mean free path is retrieved from the appropriate `CrossSections` class, and equation 4.34 is used to return the path length l_{path} for the proton in the current material.

$$l_{path} = -\lambda_{tot} ln(R_u), \tag{4.34}$$

where R_u is a uniformly distributed random number from 0 to 1.

Each `ScatteringProcess` contributes to the total cross section. The fraction of the total cross section for each process is stored in an array using the `Configure Processes()` function. For each material (or constituent for a composite), an array of fractions is created, and stored in a vector that may be accessed in the `Particle Scatter()` function when point like scattering occurs.

4.8.5.2 `EnergyLoss`

As a proton travels through a material it collides with electrons, these collisions may result in the removal of electrons, and a loss in proton energy. The interaction is defined by the Bethe-Bloch equation [87]. MERLIN offers an overloaded `EnergyLoss()` function to perform the energy loss that takes place due to ionisation in a material.

The ST-like method uses the simple Bethe-Bloch, or the stopping power, shown in equation 4.35, which describes the energy lost by a particle traversing a material with atomic number and mass Z and A [88]:

$$-\frac{dE}{dx} = Kz^2\frac{Z}{A}\frac{1}{\beta^2}\left[\frac{1}{2}ln\frac{2m_ec^2\beta^2\gamma^2T_{max}}{I_{mean}^2} - \beta^2 - \frac{\delta}{2}\right], \tag{4.35}$$

where β and γ are the relativistic kinematic variables describing the proton velocity, I_{mean} is the mean excitation energy of the target atom, K is a constant, m_e is the electron mass, c is the speed of light in a vacuum, δ is a density effect. T_{max} is the maximum transferable energy in a single collision, as defined in equation 4.36 [87]:

$$T_{max} = \frac{2m_ec^2\beta^2\gamma^2}{1 + 2\gamma\frac{m_e}{M} + \left(\frac{m_e}{M}\right)^2}, \tag{4.36}$$

where M is the mass of the incident particle. For the ST-like `EnergyLoss()` function, the value of $\frac{dE}{dx}$ is stored in the `material` class, and is used to perform the energy loss for the path length that the proton travels.

The overloaded function is a more complete treatment of the energy loss due to ionisation using the full Bethe-Bloch, shown in equation 4.37 [49]:

$$\frac{dE}{dx} = 2\pi r_e^2 m_e c^2 n_e \frac{z^2}{\beta^2}\left[ln\left(\frac{2mc^2\beta^2\gamma^2T_{max}}{I_{mean}^2}\right) - 2\beta^2 - \delta + H\right], \tag{4.37}$$

where r_e is the classical electron radius, n_e is the electron number density of the material, z is the incident particle charge, and H contains high order corrections. The higher order corrections are considered in [49]. In summary MERLIN only adds the

effects that are relevant to LHC energies; the effect due to the dielectric polarisability of solid materials, the Mott correction which is an enhancement from close collisions due to spin, and the finite size correction taking into account the size and structure of the proton. As well as these corrections, the energy spread of the outgoing proton is sampled using the Landau distribution, which is a more accurate representation of the physical effect [87].

4.8.5.3 `Straggle`

A proton travelling through a material will perform many small-angle elastic scatters from the electrons and nuclei, known as multiple Coulomb scattering (MCS), illustrated for a particle traversing a distance x through a material in Fig. 4.10.

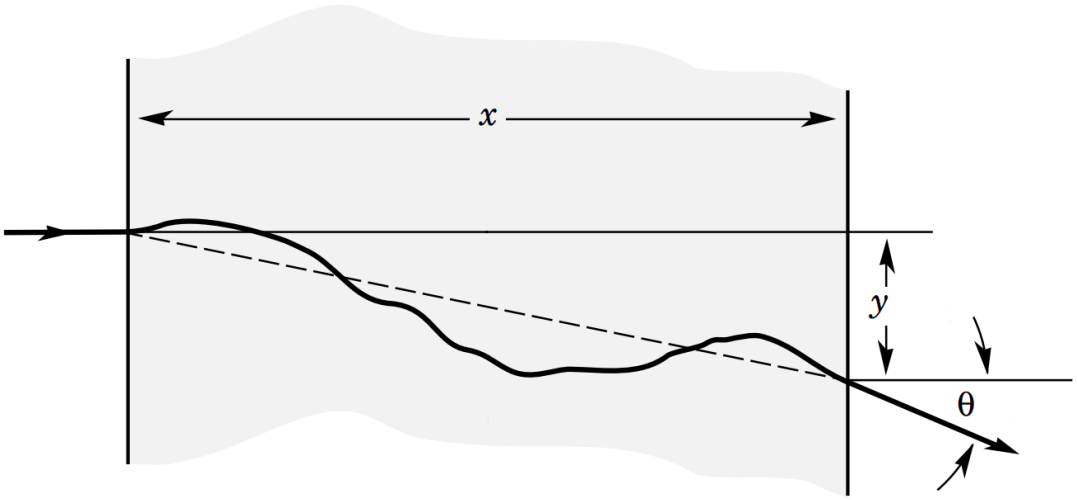

Figure 4.10: Multiple Coulomb scattering of a particle travelling through a material of thickness x. The particle undergoes a number of interactions, and leaves the material with an offset (y), and an angular kick (θ). In reality there is an offset and kick in two dimensions, only one is shown. From [90].

MERLIN and SixTrack treat MCS in the same way [49], as outlined by the PDG [87]. A Gaussian distribution is sampled to perform MCS, with an r.m.s. width of θ_0 as shown in equation 4.38:

$$\theta_0 = \frac{13.6\ MeV}{\beta cp}\sqrt{\frac{\Delta x}{\chi_0}}\left[1 + 0.088\ ln\left(\frac{\Delta x}{\chi_0}\right)\right], \tag{4.38}$$

where p is the particle momentum, χ_0 is the radiation length of the material, and

Δx is the path length taken in the material. The particle angle θ_{MCS} and offset y_{MCS} in each transverse plane are generated using equations 4.39 and 4.40 respectively:

$$\theta_{MCS} = R_u \theta_0, \tag{4.39}$$

$$y_{MCS} = R_u \Delta x \frac{\theta_0}{\sqrt{12}} + \Delta x \frac{\theta_{MCS}}{2}, \tag{4.40}$$

where R_u is a uniform random number between 0 and 1 that is re-generated at every call. In this way MERLIN simulates the MCS shown in Fig. 4.10 in both planes independently, for each path length taken in a material.

4.8.5.4 `ParticleScatter`

`ParticleScatter()` is called when a particle has travelled its path length and remains in the material, which means it will interact with a material nucleus or nucleon (*i.e.* a point-like scatter). When using a composite, by default the `CompositeMaterial::GetRandomSymbol()` is used to return the symbol of a weighted random constituent element in the composite, which is then used to select the corresponding `CrossSections` object from the map of stored `CrossSections`. However when using the SixTrack like method the `CrossSections` object for the imaginary composite atom is used, and thus the probability of each `ScatteringProcess` is that of the homogeneous mixture rather than its constituents.

Next the corresponding array of cross section fractions is called and used to randomly select and call the `ScatteringProcess::Scatter()` function which performs particle scattering. The function then returns a boolean to indicate whether or not the particle was lost in the scatter (via inelastic scattering or sufficient energy loss).

4.8.6 Note on Uncertainties

Measured total and elastic cross sections are given for proton-proton collisions in [87], which can be used to calculate the nuclear cross sections for given materials using the methods described in this chapter. From the total cross section the mean free path of a proton of given energy is calculated in that material. The mean free path gives the distance between nuclear interactions. These interactions are given a weighted probability, where the weight corresponds to their individual cross section. The cross sections therefore affect loss map simulations in two ways.

Cross sections are calculated based on models which extrapolate experimental data,

for example the MERLIN single diffractive cross section is calculated as described in [49]. Uncertainty is given from experimental data, and the fit to said data that provides the model used to extrapolate and calculate the cross section at a given energy. Depending on the amount of available data for each individual scattering type, uncertainties can be large when extrapolating to LHC energies. MERLIN takes the measured total, inelastic, and Rutherford cross sections for a material as input when available. The uncertainty on these values vary, again depending on measured data or the fit or model used to calculate them [86]. The extrapolation of cross sections to LHC energies is required, in some cases there is little data to cover the energy range of interest. See [49] for a more detailed example.

The uncertainty in the single diffractive cross section is likely to be the largest because of the lack of data. This data is used to create a model as described in [49] from which the cross section at a given energy may be obtained. The cross section at LHC energies has an uncertainty of $\approx 10\%$ - 12% [86]. In order to obtain the uncertainty in loss map simulations (*i.e.* in collimation inefficiency) numerous approaches may be taken. Propagating the cross sections uncertainties into the Monte Carlo loss map simulations is non-trivial. A numerical approach may be used, such as that in [91], where the single diffractive cross section was manually manipulated by $\pm$ 20%, and the effect on losses in specific magnets was observed. It was found that the inefficiency in dispersion suppressor dipoles immediately after the betatron collimation insertion (IR7) may change by up to 60%. In some of these magnets the variation in inefficiency was negligible. These DS magnets provide the largest inefficiency in standard loss map simulations. For machine operation an accurate estimate of this inefficiency is required These particular losses are dominated by single diffractive interactions in the collimators, therefore they are heavily dependent on this cross section. A similar study is planned for MERLIN.

4.9 Collimation Outputs

A number of output functions have been constructed to obtain useful data from collimation simulations. Details for these outputs may be found in Appendix D, some examples are given here.

The impact parameter of a proton on collimator jaws can be useful, either to verify the expected parameter when starting a simulation, or to see the influence of a process such as the hollow electron lens. The `ScatteringModel::JawImpact()` function outputs the location of collimator jaw impacts on the front face of a selected collimator.

An example for a single turn in a collimation simulation is shown in Fig. 4.11, showing the correspondence between the initial distribution (which starts immediately in front of the primary collimator), with the recorded impact co-ordinates.

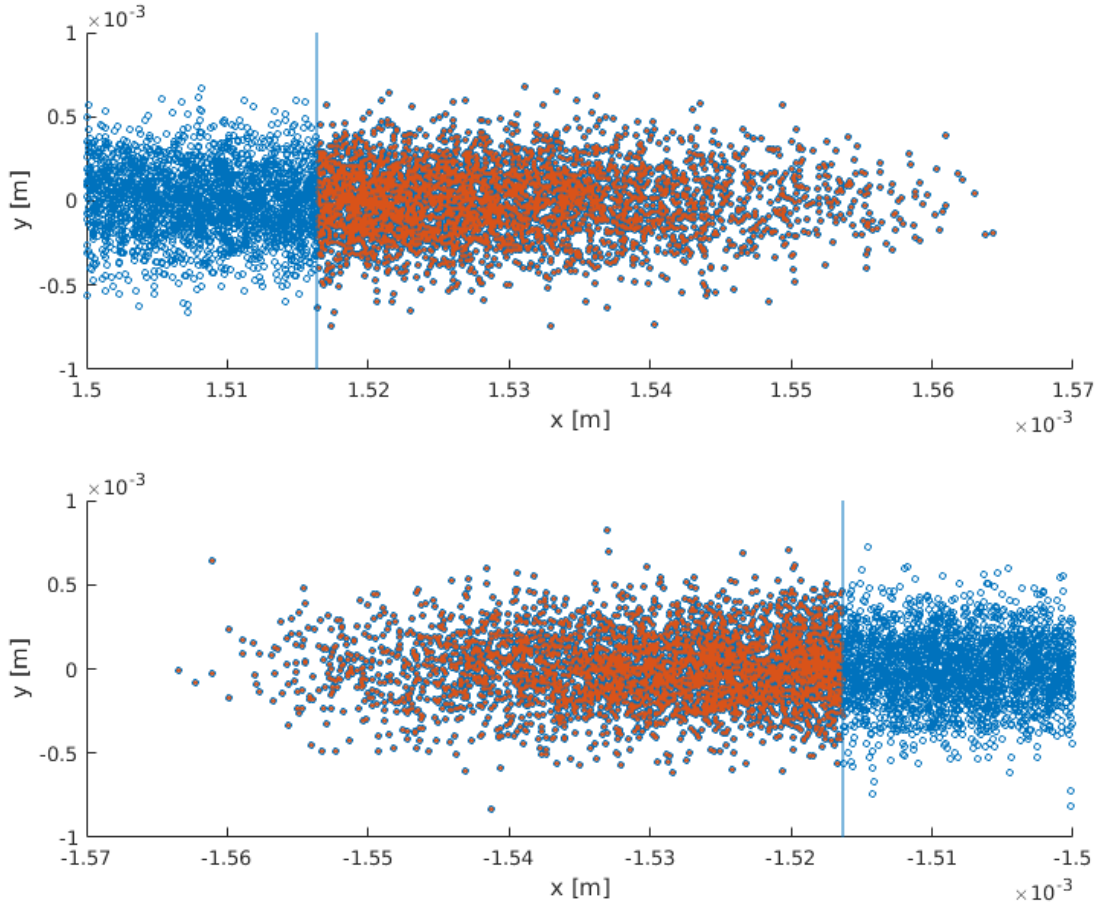

Figure 4.11: Initial distribution (blue) and impacts recorded on the primary horizontal collimator (orange) using `JawImpact`, for the positive (above) and negative (below) collimator jaws. The blue line indicates the collimator aperture, where the jaw begins. This simulation is for the 6.5 TeV LHC at flat top, using beam 2.

For debugging and visualisation it is useful to view the tracks that particles take while scattering in a collimator jaw. The `ScatteringModel::ScatterPlot()` function allows us to plot these tracks, as shown in Fig. 4.12.

We observe from Fig. 4.12 the effect of the collimation bin size. Particles undergo scattering, MCS, and ionisation energy loss for as long as they are in the collimator jaw. Even if a particle has exited a collimator jaw, the aperture check cannot take place until the particle has travelled a path length l_{path}, or at the end of the collimation bin. This

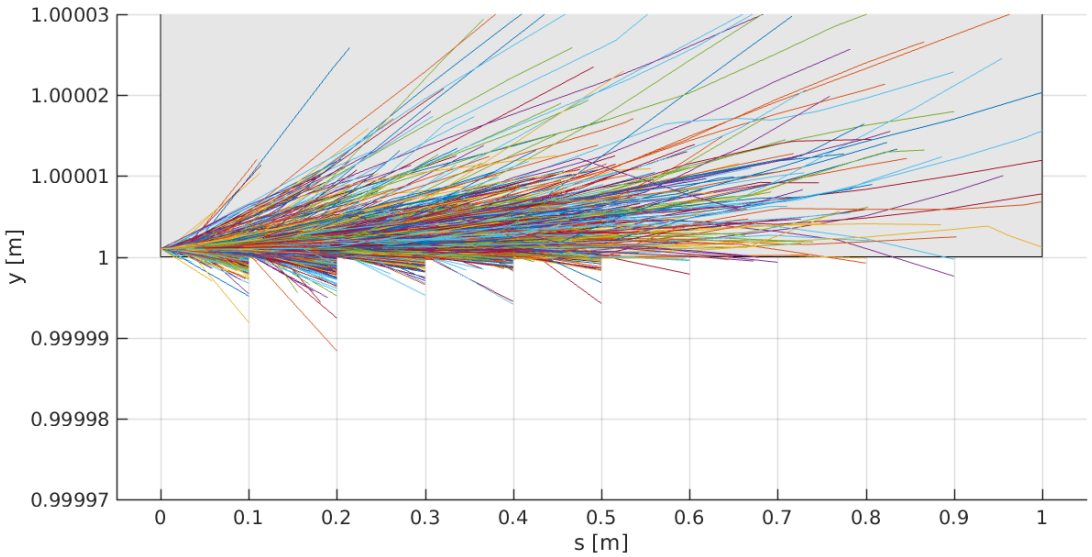

Figure 4.12: `ScatterPlot` output showing the proton tracks taken in a 1 m long copper collimator with an impact paramter of 1 μm in the y plane, using a 10 cm collimation bin size. The grey area indicates the collimator jaw, and the particles are not tracked by this output if they exit the collimator jaw.

is evident as particle tracks abruptly stop at 10 *cm* intervals in the figure. By reducing the bin size, a small path length is forced, and computation time will increase, however by using a larger bin size protons may undergo significantly more bulk scattering (MCS and ionisation energy loss) when they have in fact already left the collimator jaw. The 10 *cm* bin size that is used by default allows regular aperture checks without enforcing too small a path length, or compromising the condition of protons that return to the bunch after undergoing scattering in a collimator jaw.

The location of an inelastic interaction along a collimator jaw may be output using the `ScatteringModel::JawInelastic()` function. MERLIN assumes that a proton is lost if an inelastic interaction occurs. This provides a necessary comparison tool to observe the effect of different collimator materials, an example histogram of the distribution of losses in a given collimator is shown in Fig. 4.13.

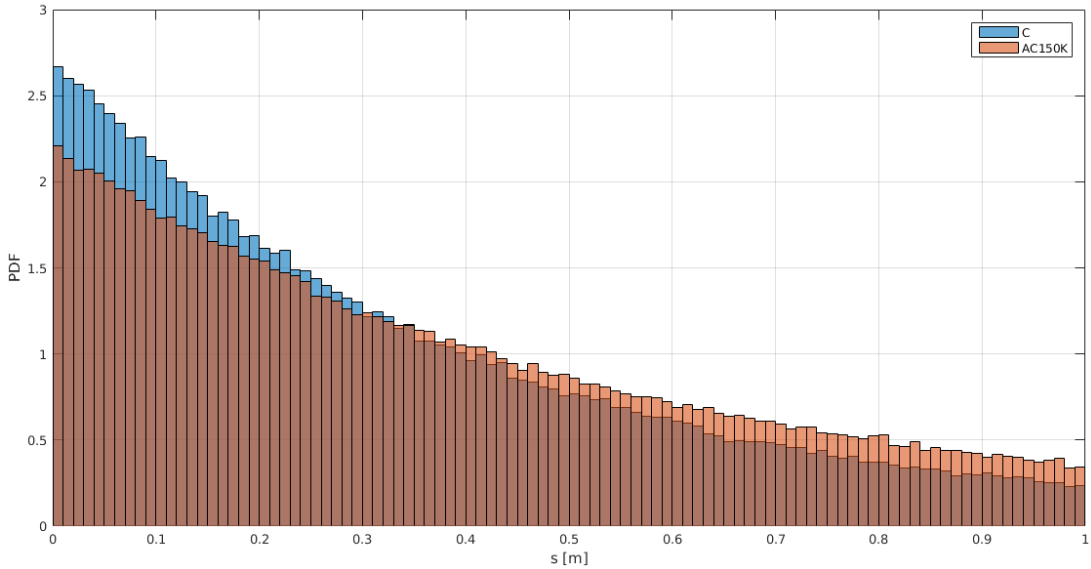

Figure 4.13: Inelastic proton interactions (proton losses) in a secondary collimator in the nominal LHC using the `JawInelastic` output. Comparing losses in pure carbon (blue) with CFC AC150K (orange).

4.10 Hollow Electron Lens Process

The hollow electron lens (HEL) process provides the means to simulate the effect of a HEL on a particle beam. This process makes a number of assumptions (detailed in Chapter 3), and treats the HEL only as a collimation enhancer, simulating the small angle kick that a proton interacting with the HEL EM field receives. The `SetCurrentComponent()` function checks that the current `AcceleratorComponent` is a `HollowElectronLens`, if not the process exits.

In order to insert the HEL into the lattice as a thin (zero length) element, the process requires the user to specify the active length. The HEL is treated as a drift tube by MERLIN as mentioned previously, however even if the HEL has a length in the lattice, its active length must be specified by the user using one of the provided `HollowELensProcess` constructors, or the `SetEffectiveLength()` function.

The basic constructor allows the user to specify the HEL current, electron Lorentz β, and proton beam rigidity. Overloaded constructors add the effective length, and the radii of the HEL. The radii may be set in two ways, in a similar fashion to the setting of collimator half gaps.

The HEL process records its own turn value for the simulation, this is necessary for the turnskip and AC modes of current modulating operation. The user should note that this turn value is incremented every time the process is called, thus assuming a single HEL in the lattice.

A complete example of the method of creating the HEL process may be found in Appendix D.

4.10.1 Radii Setting

The first method using the function `SetRadii()` sets the radii in units of m, and does not take into account the beam envelope or the closed orbit.

The second method uses the `SetRadiiSigma()` function, which takes the radii in units of beam σ, the transverse beam emittances, and the `LatticeFunctionTable` for the lattice. The function iterates through the `LatticeFunctionTable`, finds the position of the HEL in the lattice, and uses the β function in the horizontal x plane to set the HEL inner and outer radii, R_{min} and R_{max} respectively. If the beam is not round at this position (*i.e.* $\beta_x \neq \beta_y$), as the HEL beam is round there will be a mismatch of the HEL overlap with the beam halo and core. The overloaded `SetRadiiSigma()` simply adds the phase advance to the list of details output to screen when computing the matched radii, using the new `PhaseAdvance` class.

As in SixTrack, two functions are provided to include the radial dependence of the EM field generated by the HEL beam. The 'perfect' profile is the piecewise function given in equation 3.6, and may be set using the `SetPerfectProfile()` function. The second profile is a parameterisation of the measured prototype cathode [92], and may be selected using the `SetRadialProfile()` function. This parameterisation is given for the Tevatron-like HEL parameters, and so when using the LHC parameters an empirical adjustment must be made to produce the expected profile, this is switched on using the `SetLHCRadialProfile()` function.

Two functions exist to output the HEL profile and footprint. `OutputProfile()` outputs the value of the HEL kick between a minimum and maximum value (in units of σ) of the positive x co-ordinate, using 1000 points of reference equally spaced between these. For example a user may specify a range of 0 - 10 σ, and the HEL profile will be output every 0.01 σ in x. The output file columns are described in Table 4.10. `OutputFootptint()` creates a sample of points in transverse space between $\pm\frac{3}{2}R_{max}$, and performs a check to find whether or not these points hit the HEL beam or not. If so they are included in the output file which, when plotted, will produce the footprint of the HEL. The footprint output is useful when observing the effect of novel geometrical operation modes of the HEL. Table 4.11 indicates the columns for `OutputFootprint()`.

r $[\sigma]$	$Kick_{radial}$	$Kick_{perfect}$	$\lvert Kick_{radial}\rvert$	$\lvert Kick_{perfect}\rvert$

Table 4.10: Column headings for the `HollowELensProcess::OutputProfile()` file.

x $[m]$	y $[m]$	r $[m]$

Table 4.11: Column headings for the `HollowELensProcess::OutputFootprint()` file.

4.10.2 Current Modulation Modes

Four current modulating operation modes are available in MERLIN, these are detailed in Chapter 3, and may be selected using the `SetOpMode()` function which takes an enumerator (`DC, AC, Diffusive,` or `Turnskip`). The DC and Diffusive modes do not require any further information, however the turnskip and AC modes require further functions to specify certain variables. The number of turns defining the turnskip operation is set using the `SetTurnskip()` function, which takes an integer.

The AC mode requires a number of variables, these are set using the `SetAC()`

function, which takes the machine tune ν, the tune range Δ_ν (the function sweeps between the tune $\pm\Delta_\nu$), the tune variation per step, the turns corresponding to a single step, and an integer multiplier m. The AC mode works by attempting to modulate the HEL current in resonance with the machine tune ν, for the full derivation of this method see [68]. An operating tune (or frequency) ν_{op} is defined using equation 4.41:

$$\nu_{op} = \nu_{min} + w \cdot \delta_{nu}, \tag{4.41}$$

where ν_{min} is the minimum tune of the specified range, w is the step, and δ_{nu} is the tune step which is taken at a user specified number of turns. ν_{op} is used to define a harmonic frequency ϕ using equation 4.42:

$$\phi = m \cdot T \cdot 2\pi\nu_{op}, \tag{4.42}$$

where T is the turn, and m is an integer multiplier. The HEL kick θ is then modulated using equation 4.43:

$$\theta = \theta \cdot \frac{1}{2}(1 + cos(\phi)). \tag{4.43}$$

These operation modes are implemented in the `DoProcess()` function, where a switch differentiates between the four current modulating operation modes.

4.10.3 Geometrical Operation Modes

MERLIN contains four novel geometrical operation modes, these and the current modulating modes are not mutually exclusive. Whereas a current modulating operation mode must be specified (or set to DC by default), a geometrical operation mode does not need to be used.

These modes are described in Chapter 3. One may be selected using the appropriate function from: `SetEllipticalMatching()`, `SetHulaElliptical()`, `SetPogo Elliptical()`, `SetCloseHulaElliptical()`. All modes attempt to improve the operation of the round HEL beam on a non-round proton beam by translating the HEL and adjusting the radii appropriately. All modes except the elliptical matching are dynamic, and assume that there is only a single HEL in the lattice when performing geometrical adjustments.

In the `DoProcess()` function conditional statements are used to implement the hula, close hula, and pogo geometries using adjustment functions. These functions use the elliptical adjustment as a basis, and include the stepping of the HEL footprint in

real space for dynamic modes on a turn by turn basis. As well as this adjustment, the flags engaged when a novel geometrical mode is initialised, are used as conditionals to decide whether or not the offsets are included in the particle angle calculation, in order to take into account the correct kick direction.

4.10.4 Kick Calculation

The maximum HEL kick is given by equation 3.5. Appendix B explains that the relative directions of the HEL and machine beams will modify the net force that provides the kick due to the addition of electric and magnetic field components. The `SetElectron Direction()` function allows the user to modify this net force by changing the direction of the HEL beam. In most cases however we only consider the case where the kick is larger, and the electric and magnetic components sum.

The `CalcThetaMax()` function is called from both kick calculation methods, it takes into account the electron direction, and uses equation 3.5 to return the maximum kick that a proton may be given due to the HEL. Two functions exist to provide the actual kick given to a proton: `CalcKickSimple()` is called when the user has selected the perfect HEL profile, and `CalcKickRadial()` is called when the user has requested the measured profile. Both of these functions take a particle, calculate its position relative to the HEL using offset variables for each transverse plane that are set in the geometrical functions, calls the `CalcThetaMax()` function, and returns the kick for the individual particle. This kick is then given to the particle using equation 4.44.

$$x' = x' + \theta \cdot cos(\psi), \qquad y' = y' + \theta \cdot sin(\psi), \tag{4.44}$$

where ψ is the particle angle as defined by equation 4.45:

$$\psi = tan^{-1} \left(\frac{y - y_{shift}}{x - x_{shift}} \right), \tag{4.45}$$

where y_{shift} and x_{shift} are the variables used to take into account any translation of the HEL centre with respect to the beam centre.

The HEL process takes place in the `DoProcess()` function, which uses a switch to perform the current modulation, and other functions to perform any geometrical translation of the HEL at each turn. `DoProcess()` then iterates through the particle bunch, calculates the HEL kick for each particle - zero for a particle that does not interact with the HEL - and applies the kick to the particle.

4.11 Summary

In this chapter we have detailed the parts of MERLIN that have been added, updated, merged, and optimised in MERLIN 5.01. The user may define their accelerator in the form of an `AcceleratorModel`, using the `MADInterface` class to read a standard MADX TFS table. Further input files may be read by the `ApertureConfiguration` class to define the apertures of the accelerator, and the `CollimatorDatabase` to set up collimators. The user may then calculate the lattice functions of the accelerator, and in turn use these to define a beam, which leads to the construction of a `ParticleBunch` that is matched to the accelerator at the desired position. A particle tracker may be constructed selecting from either `TRANSPORT` or `SYMPLECTIC` integrator sets, and physics processes may be attached to it. Finally the user may run the tracking simulation, and create outputs using a myriad of existing output functions, or define their own. This standard MERLIN code flow is shown in Fig. 4.14, which is an example given for a collimation simulation.

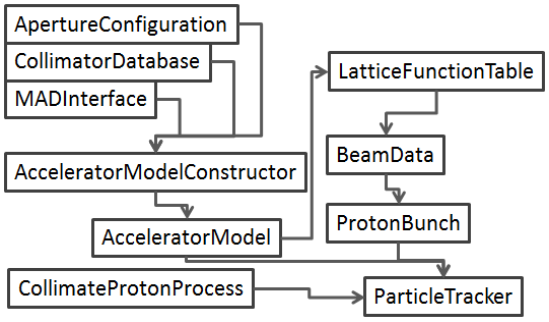

Figure 4.14: The code flow of a typical collimation simulation using MERLIN 5.01. First the user must create the `AcceleratorModel`, which in this case is done using seperate input files for the apertures, elements, and collimators, and passing them through the respective parsing classes. The lattice functions are then calculated using this `AcceleratorModel`, from which the initial bunch distribution may be generated and matched to any selected 'injection' position (*i.e.* where the bunch is tracked from). A particle tracker is created, and physics processes are added to it, in this case only the collimation process. The bunch is also given to the tracker, which then performs all tracking and user selected as well as default output.

5

MERLIN Validation

5.1 Lattice Functions

We perform a comparison of the lattice functions in MERLIN (calculated with `LatticeFunctionTable` and `Dispersion` classes) and those calculated in MADX, in order to validate the MERLIN calculations. Similarity is essential as they are a direct representation of the magnetic fields of the accelerator lattice. An agreement indicates that MERLIN and MADX are both simulating same machine. In the LHC case this machine is a 26.67 km long synchrotron and so any error in its representation will have severe consequences on the reliability of simulations. MADX has been developed for over a decade, it is well maintained, and widely accepted as accurate and reliable, thus agreement with MADX implies reliability in terms of the accelerator description. As stated in Chapter 4, the calculations are performed in 4 dimensions in both codes. Beta functions are compared in Fig. 5.1 and Fig. 5.2, dispersion in Fig. 5.3, and the closed orbit in Fig. 5.4, all for the 6.5 TeV LHC using beam 2 at flat top, which is the period after acceleration to top energy, and before the squeeze to reduce β^* at the collision IPs.

We observe excellent agreement between MERLIN and MADX in the calculation of the lattice functions, the dispersion and the closed orbit. We note that the `Dispersion` class outputs the dispersion at larger intervals than MADX, leading to a step-like plot instead of a smooth interpolation.

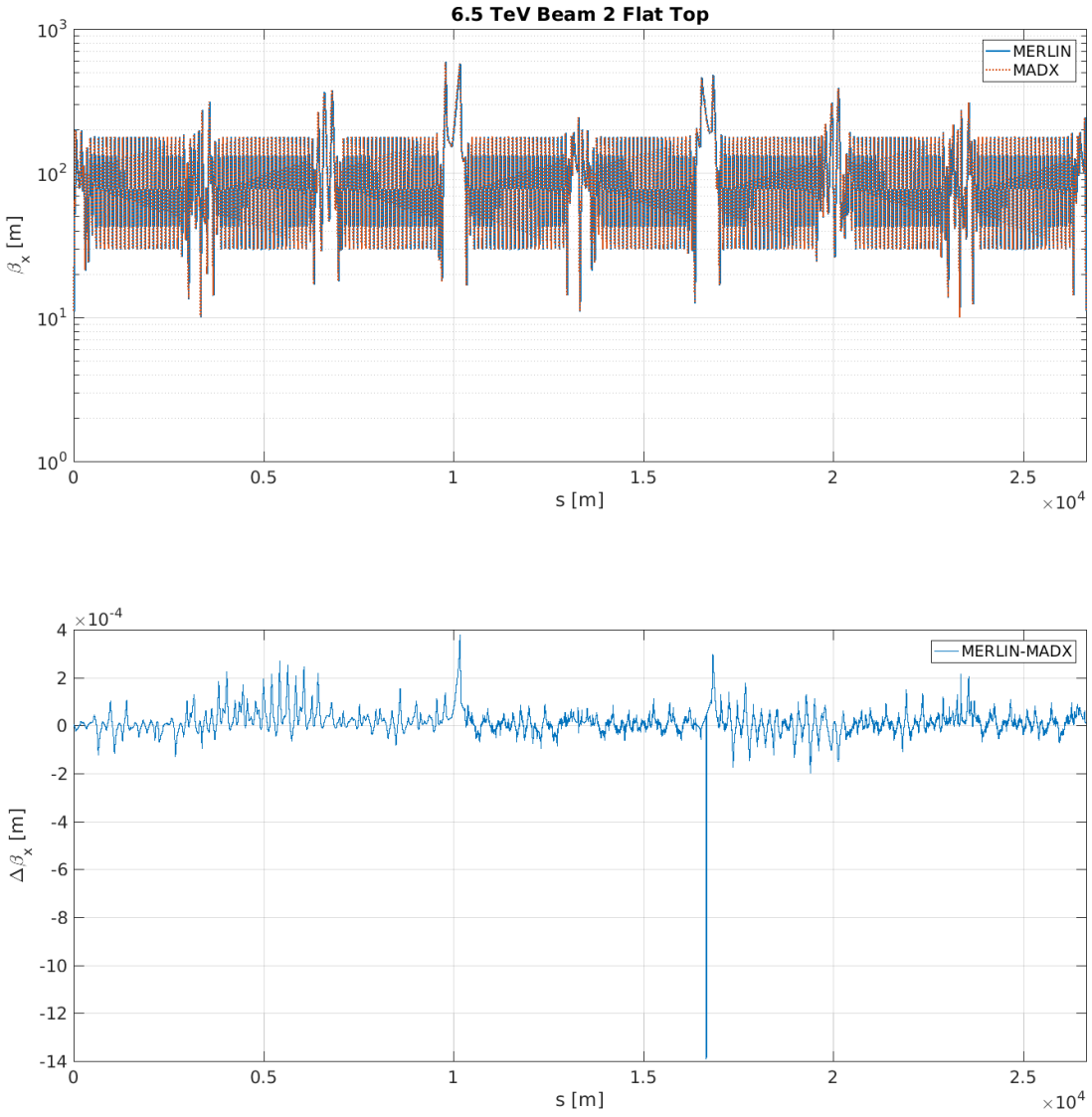

Figure 5.1: β_x function for beam 2 of the 6.5 TeV LHC at flat top, MERLIN (blue) is compared to MADX (orange) in the top plot. The bottom plot shows the difference between MERLIN and MADX, which is made using an interpolation algorithm. The difference is negligible.

5.2 Apertures

The apertures of the machine are set using a separate input file. MERLIN's interpolated aperture in the horizontal plane is shown in Fig. 5.5, and compared to a linear interpolation between the points specified in the input TFS file generated by MADX.

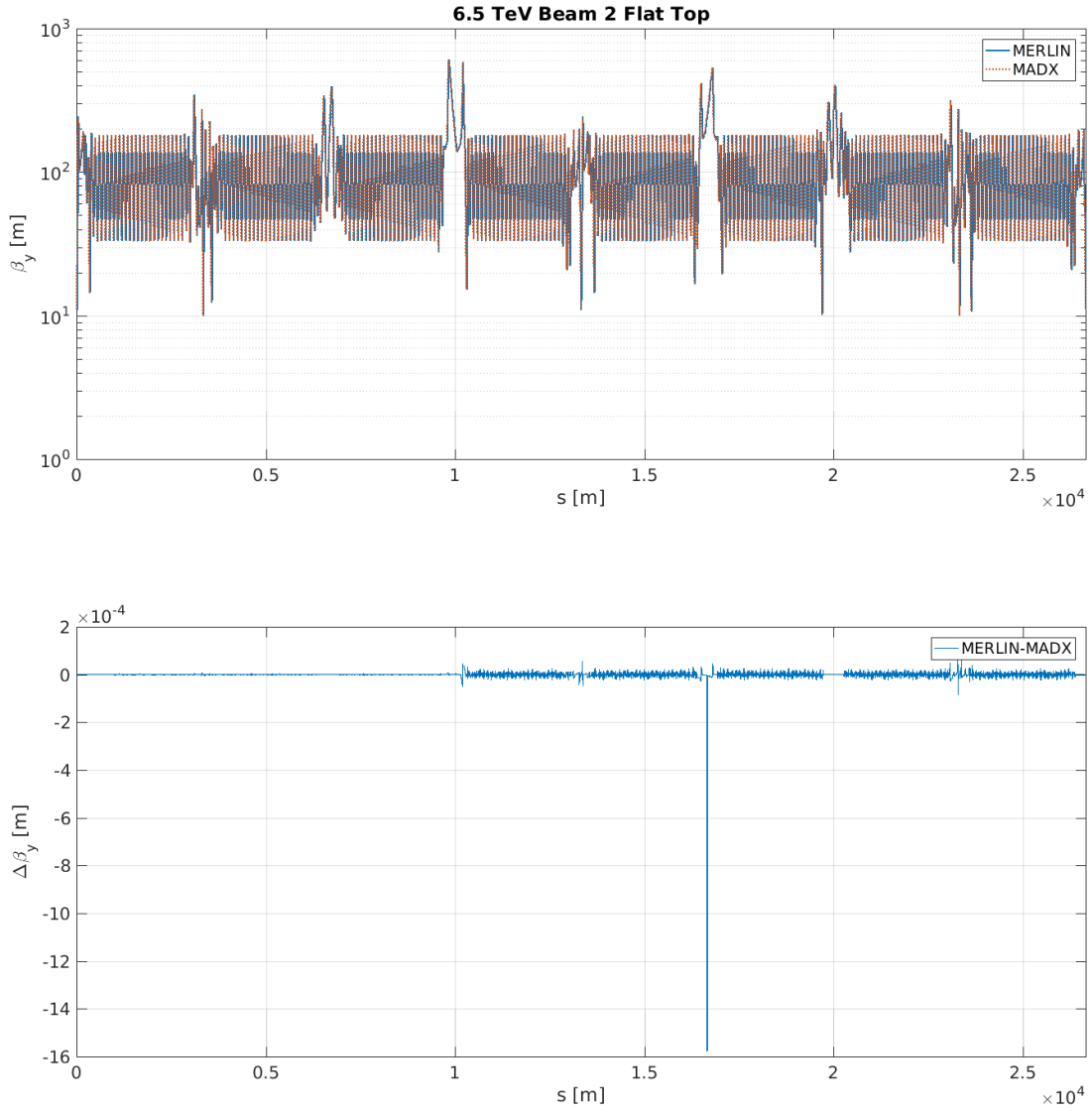

Figure 5.2: β_y function for beam 2 of the 6.5 TeV LHC at flat top, MERLIN (blue) is compared to MADX (orange) in the top plot. The bottom plot shows the difference between MERLIN and MADX, which is made using an interpolation algorithm. The difference is negligible.

The overall agreement is good, with a few small differences that are deemed insignificant. These may be because of zero length elements or the linear interpolation of the plot. As mentioned in Chapter 4, MERLIN's aperture configuration will soon be updated to take these into account.

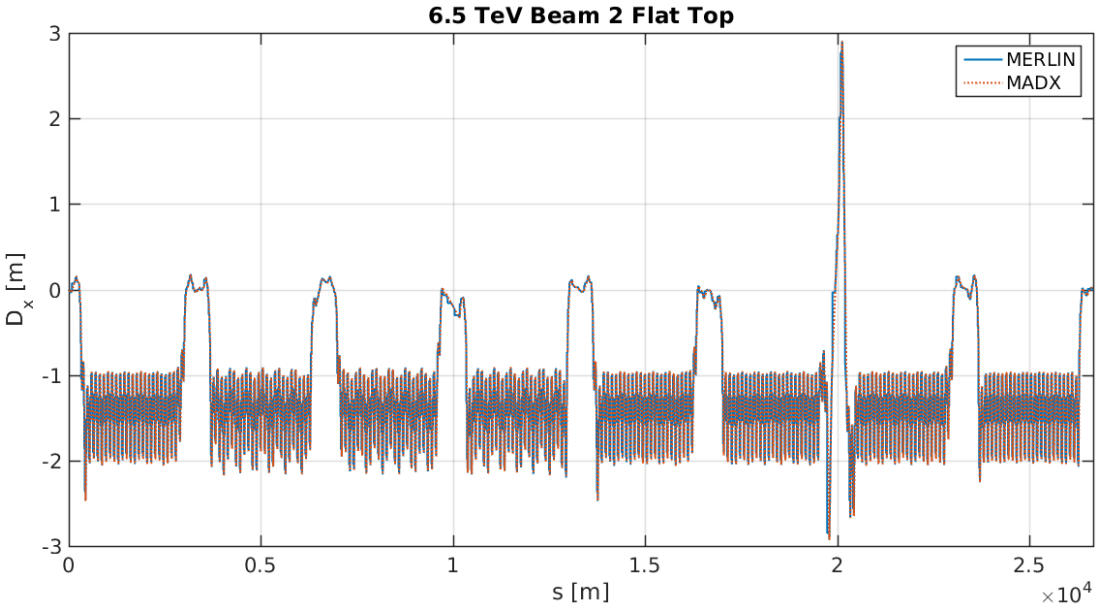

Figure 5.3: Dispersion D_x function for beam 2 of the 6.5 TeV LHC at flat top, MERLIN (blue) is compared to MADX (orange).

5.3 Tracking

As mentioned in Chapters 2 and 4, the MERLIN `SYMPLECTIC` integrator preserves the actions of the system, whereas the `TRANSPORT` method is not symplectic, and for a large number of turns will become inaccurate due to small numerical instabilities caused by the integration method. This is observed in Fig. 5.6, where particles tracked with the `TRANSPORT` integrators are perturbed from stability after a large number of turns.

We also observe this to some extent in δ vs ct phase space, as shown in Fig. 5.7. We observe the slight difference in RF structure integrators at large amplitudes.

The symplectic condition is $JSJ^T = S$, where S is antisymmetric matrix and J is the Jacobian. As the determinant of the transpose is equal to the determinant of the original matrix, the symplectic condition for a transfer map is shown in equation 5.1:

$$JSJ^T \equiv Det(J)^2 = 1 \Rightarrow Det(J) = \pm 1. \tag{5.1}$$

The effect of each individual element on a particle may be described using the transfer matrix of that element. We may combine these matrices in the correct sequence to provide a matrix that represents the transformation of a particle travelling through

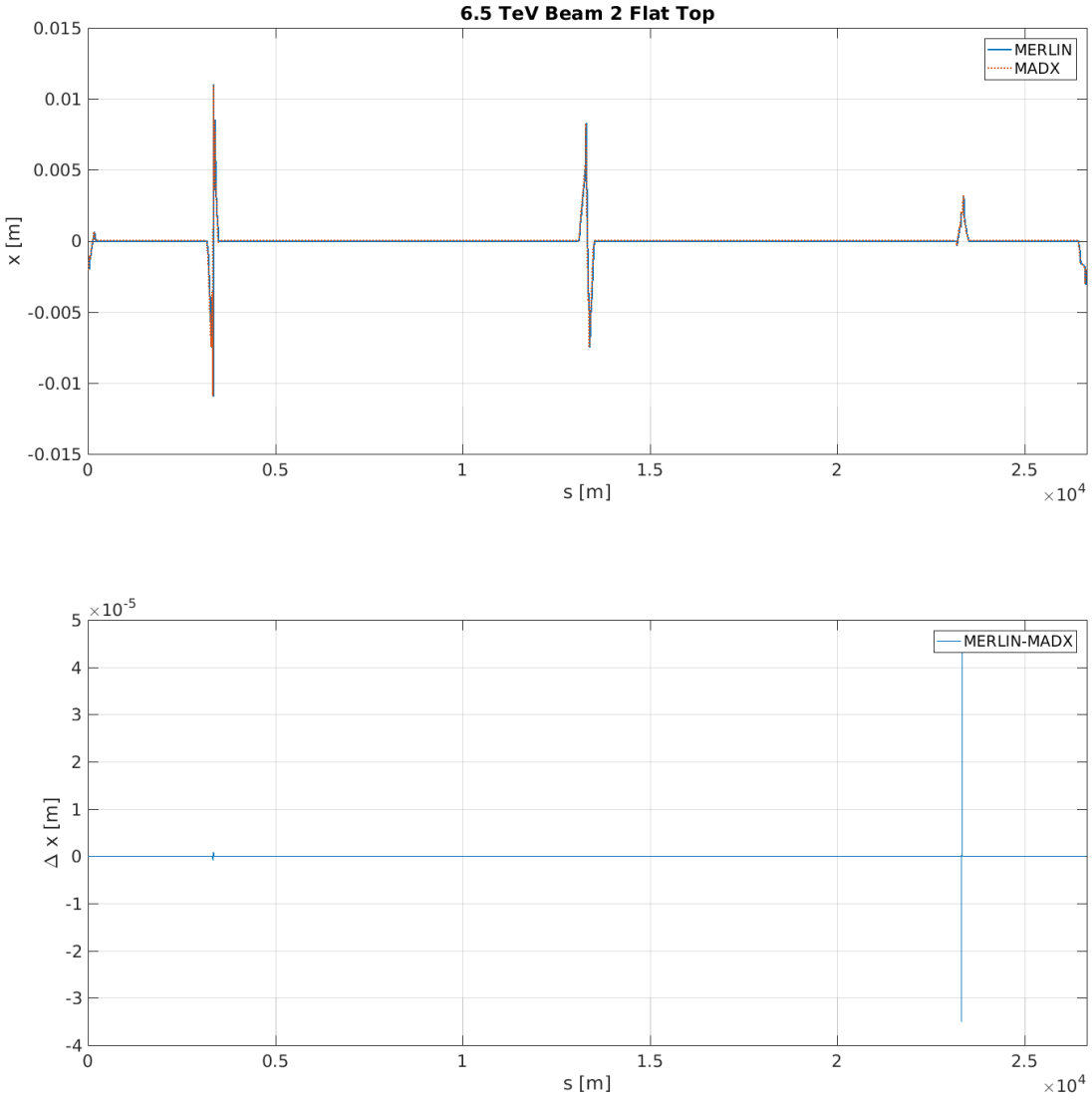

Figure 5.4: Closed orbit in the transverse x plane for beam 2 of the 6.5 TeV LHC at flat top, MERLIN (blue) is compared to MADX (orange) in the top plot. The bottom plot shows the difference between MERLIN and MADX, which is made using an interpolation algorithm. The difference is negligible.

a single turn of the accelerator. In a linear approximation this is the one-turn-map (OTM).

Instead of calculating the determinant of each individual transfer matrix, we may find the determinant of the one-turn-map $|DET(OTM)|$, to compare `SYMPLECTIC` and `TRANSPORT` tracking. This is done as a function of transverse horizontal displacement

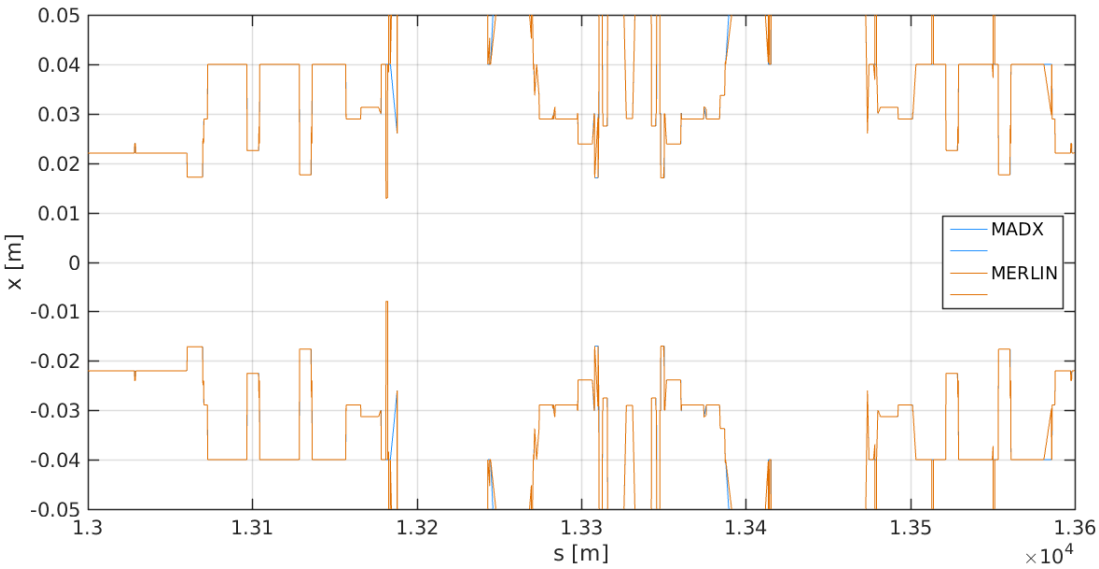

Figure 5.5: Apertures in the transverse x plane for beam 2 of the 6.5 TeV LHC at flat top at IP5, MERLIN (blue) is compared to MADX (orange).

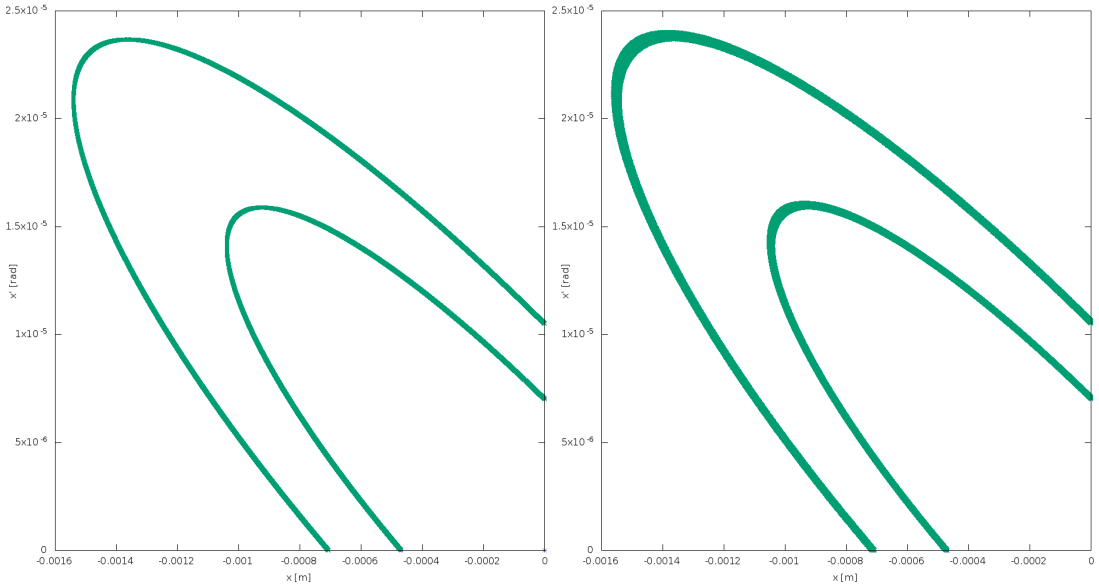

Figure 5.6: Poincaré section in xx' phase space for two arbitrarily chosen particles in the nominal LHC for 10^6 turns. The left hand plot shows tracks from `SYMPLECTIC` tracking, the right hand plot shows tracks from `TRANSPORT` tracking.

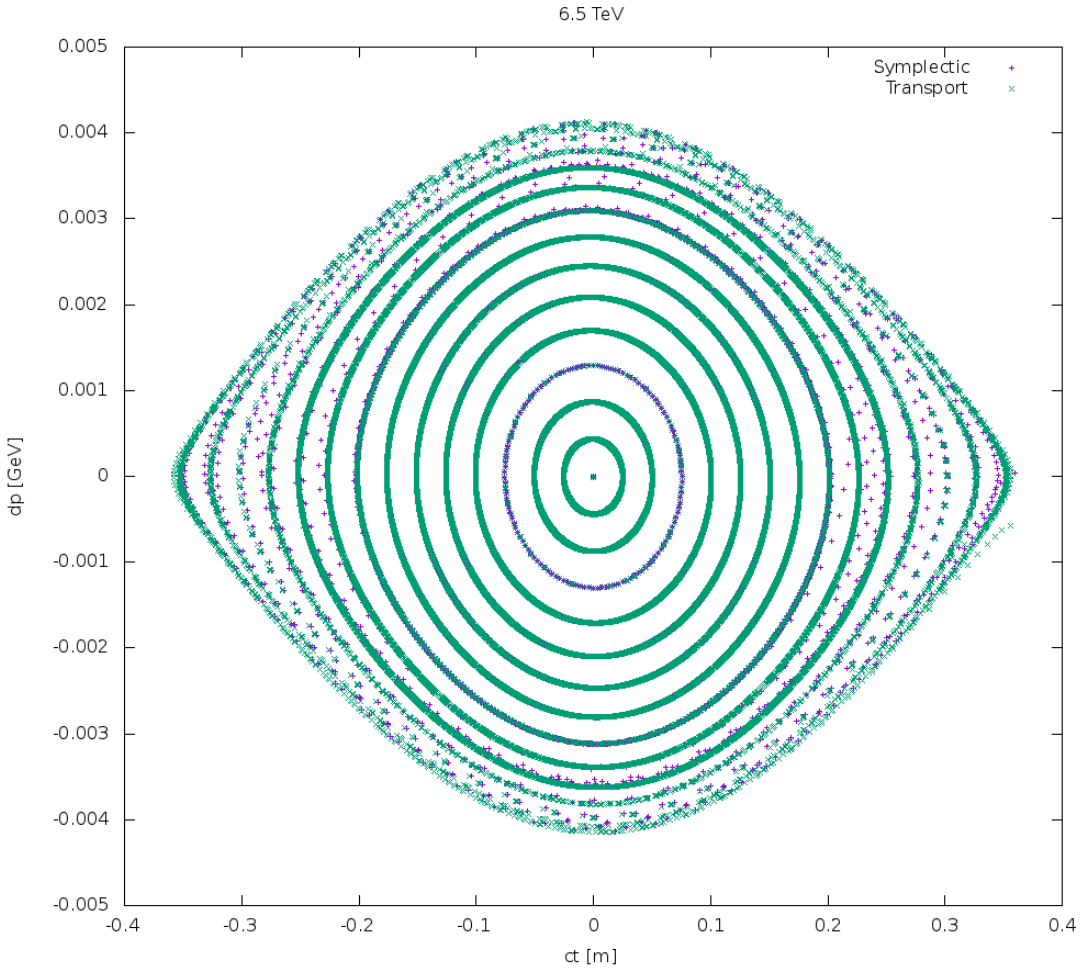

Figure 5.7: Poincaré section in ct vs δ phase space. Purple tracks show `SYMPLECTIC` tracking, green tracks show `TRANSPORT` tracking.

x in Fig. 5.8, and as a function of longitudinal displacement ct in Fig. 5.9. In these plots, an initial pencil distribution (one with all initial co-ordinates set to 0) is used with either a spread in x or a spread in ct.

We observe that deviation of $|DET(OTM)|$ from 1 for the `SYMPLECTIC` tracker is smaller than for the `TRANSPORT` tracker in both cases. For transverse displacement the deviation from one increases as the particle moves further from the closed orbit, however this is not the case for longitudinal displacement. This is due to the fact that the magnetic fields for most elements vary as a function of transverse displacement, whereas a particle arriving earlier or later will see the same field.

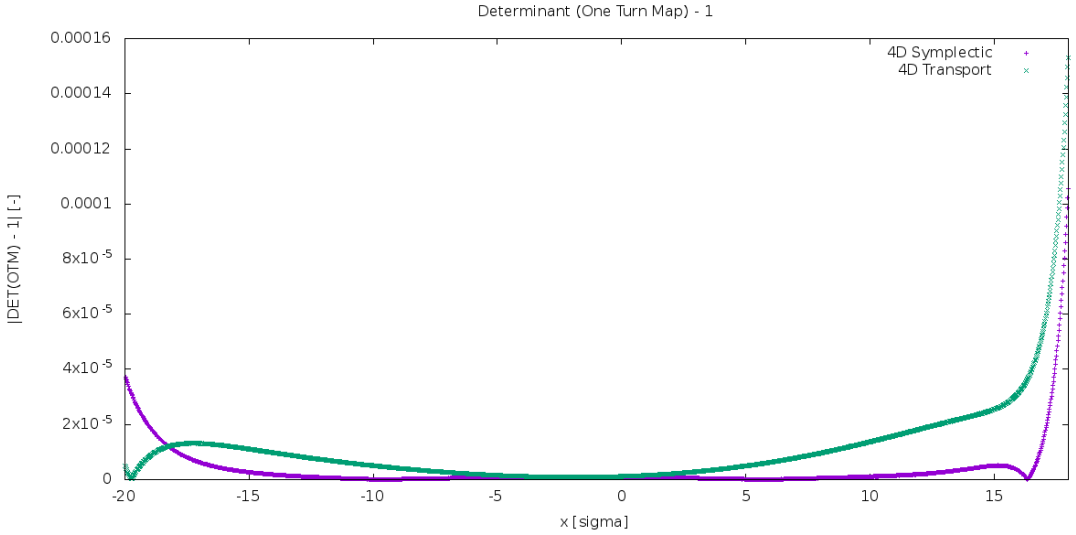

Figure 5.8: The determinant of the one turn map minus 1, as a function of initial transverse displacement x. Purple tracks show `SYMPLECTIC` tracking, green tracks show `TRANSPORT` tracking.

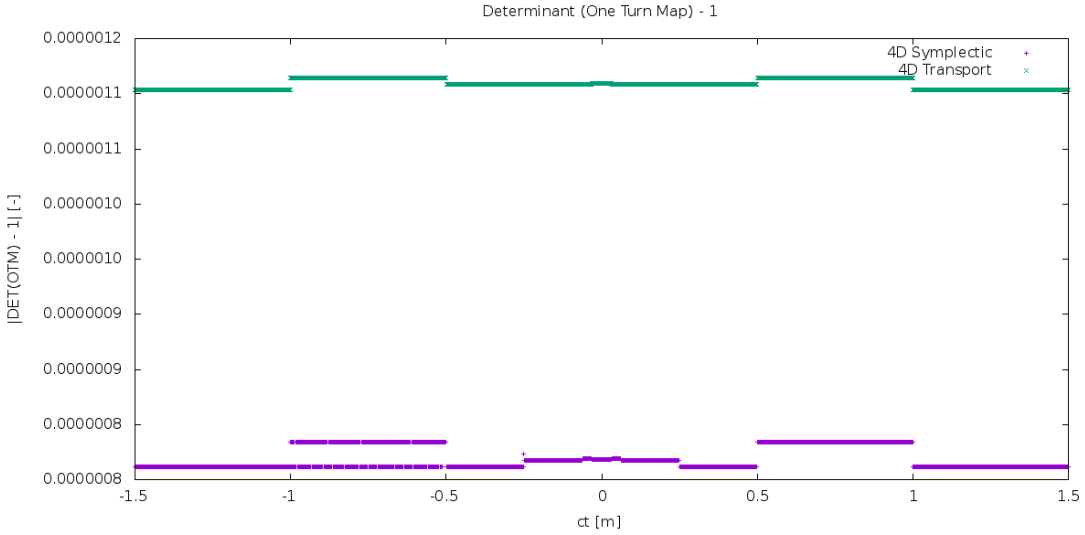

Figure 5.9: The determinant of the one turn map minus 1, as a function of initial longitudinal displacement ct. Purple tracks show `SYMPLECTIC` tracking, green tracks show `TRANSPORT` tracking.

It is important to note that there is no standard acceptable range for this figure of merit, it is merely used to compare the two integrator sets, and suggest that the

`SYMPLECTIC` method offers some advantage to the `TRANSPORT` method.

5.4 Scattering in Collimator Materials

The constituents of the most commonly used composite materials for the LHC and HL-LHC collimator jaws are shown in Table 5.1. Table 5.2 compares their cross sections and properties. CFC AC150K is not shown in Table 5.2 as it is treated as a pure material, and is identical in both SixTrack and MERLIN.

Material	**Z**	**A** $[\frac{g}{mol}]$	ρ $[\frac{g}{cm^3}]$	σ_{el} $[\frac{MS}{m}]$	Atomic Content [%]
CFC	6	12.01	1.67	0.14	100 C
MoGr	6.653	13.532	2.5	1	2.7 Mo_2C, 97.3 C
CuCD	11.898	25.238	5.4	12.6	25.7 Cu, 73.3 CD, 1 B
Glidcop	28.823	63.149	8.93	53.8	99.1 Cu, 0.9 Al_2O_3
Inermet180	67.657	166.68	18	8.6	86.1 W, 9.9 Ni, 4 Cu

Table 5.1: Collimator jaw materials and their corresponding atomic contents and properties [45].

5.4.1 Proton Scattering

A test case was used to compare the proton scattering in MERLIN with that of other codes. The simulation involves impacting a pencil beam on a vertical collimator jaw 0.5 m long, at an impact parameter of 1 μm. The horizontal output distribution is plotted and compared with the the same test case performed with FLUKA [19], Struct [94] and K2 [81]. We note that the test case performed using these codes was done a number of years ago in [94], and it is likely that the models used have since been updated. This comparison remains valid as any update will only likely have small effect on the distributions exiting the collimator. The length of the collimator ensures that the scattering is not dominated by a single point-like process.

We may compare the effect of scattering between the loss map version and version 5.01 of MERLIN in Fig. 5.10. We observe that there is a small difference in the transverse position spread and peak of the exiting proton beam. Performing the same comparison using ST-like scattering in both versions of MERLIN in Fig. 5.11, the discrepancy is similar. In these figures we observe good agreement with the data from [94], noting that the transverse distributions when using ST-like scattering in MERLIN 5.01 are similar to the K2 routine on which SixTrack's proton scattering is based. It is

Property	MoGr	CuCD	Glidcop	Inermet180
$\sigma_{pN\ tot\ ref}$				
MERLIN	0.3545	0.572	1.246	2.546
SixTrack	0.362	0.572	1.246	2.548
$\sigma_{pN\ I\ ref}$				
MERLIN	0.244	0.370	0.765	1.525
SixTrack	0.247	0.370	0.765	1.473
$\sigma_{pN\ R\ ref}$				
MERLIN	$11.9{\cdot}10^{-5}$	0.000449	0.00151	0.006807
SixTrack	$9.4{\cdot}10^{-5}$	0.000279	0.001385	0.005737
$\sigma_{pN\ tot}$				
MERLIN	0.3827	0.6068	1.293	2.611
$\sigma_{pN\ E}$				
MERLIN	0.07459	0.1584	0.4217	0.939
$\sigma_{pn\ E}$				
MERLIN	0.04487	0.05536	0.07516	0.1039
$\sigma_{pn\ SD}$				
MERLIN	0.001879	0.02318	0.03148	0.0435
$\lambda_{tot}\ [m]$				
MERLIN	0.2301	0.1259	0.0892	0.0577
SixTrack	0.2484	0.1356	0.0942	0.0603

Table 5.2: Cross sections and total mean free path of composite materials in MERLIN and SixTrack [93], all values are given in barns unless specified. Subscripts: $\_pN$ proton nuclear, $\_pn$ proton nucleon, $\_ref$ reference, $\_tot$ total, $\_I$ inelastic, $\_E$ elastic, $\_sd$ single diffractive, $\_R$ Rutherford. λ_{tot} is the mean free path.

also clear that the energy loss when using MERLIN scattering has a smaller cut-off. Figure 5.12 compares MERLIN and ST-like scattering in MERLIN 5.01. From this the difference in the energy loss distribution is clear, as well as the slightly larger transverse spread when using MERLIN scattering.

Thus we may conclude that MERLIN is in good agreement with existing codes for this test case, with the ST-like scattering within MERLIN noticeably similar to the K2 scattering which forms the basis of SixTrack proton scattering. The most prominent difference is the energy loss distribution when using MERLIN scattering, which will be investigated in the following subsections, and is due to the advanced single diffractive scattering routine.

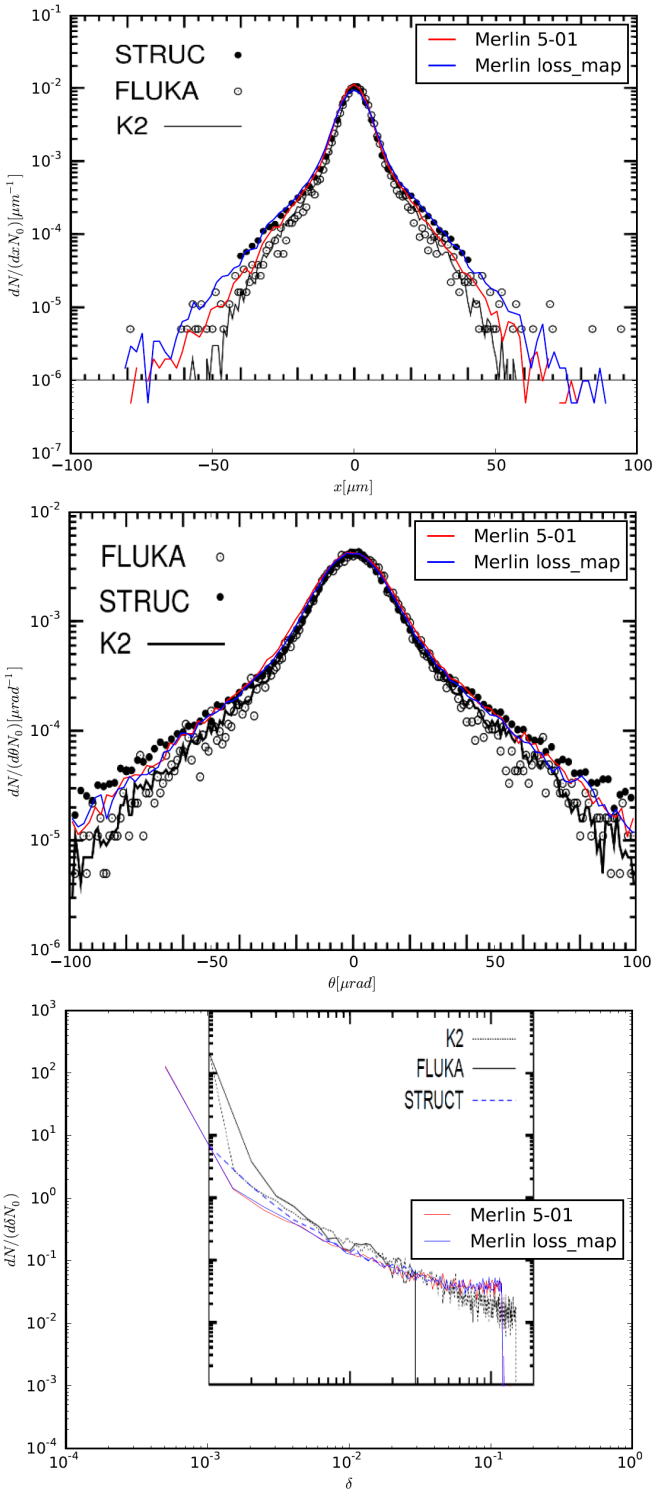

Figure 5.10: Scattering probabilities for a 7 TeV proton impacting on a 0.5 m long Cu jaw. Change in position (top), angle (middle) and energy (bottom) are compared for MERLIN 5-01 (red) and the loss map version of MERLIN (blue). Both cases use MERLIN scattering. The data is overlaid onto that from the same study performed using other codes in [94]. We observe a difference between MERLIN scattering and the other codes in the δ distribution tail, the reason for this is clarified in Section 5.4.3.

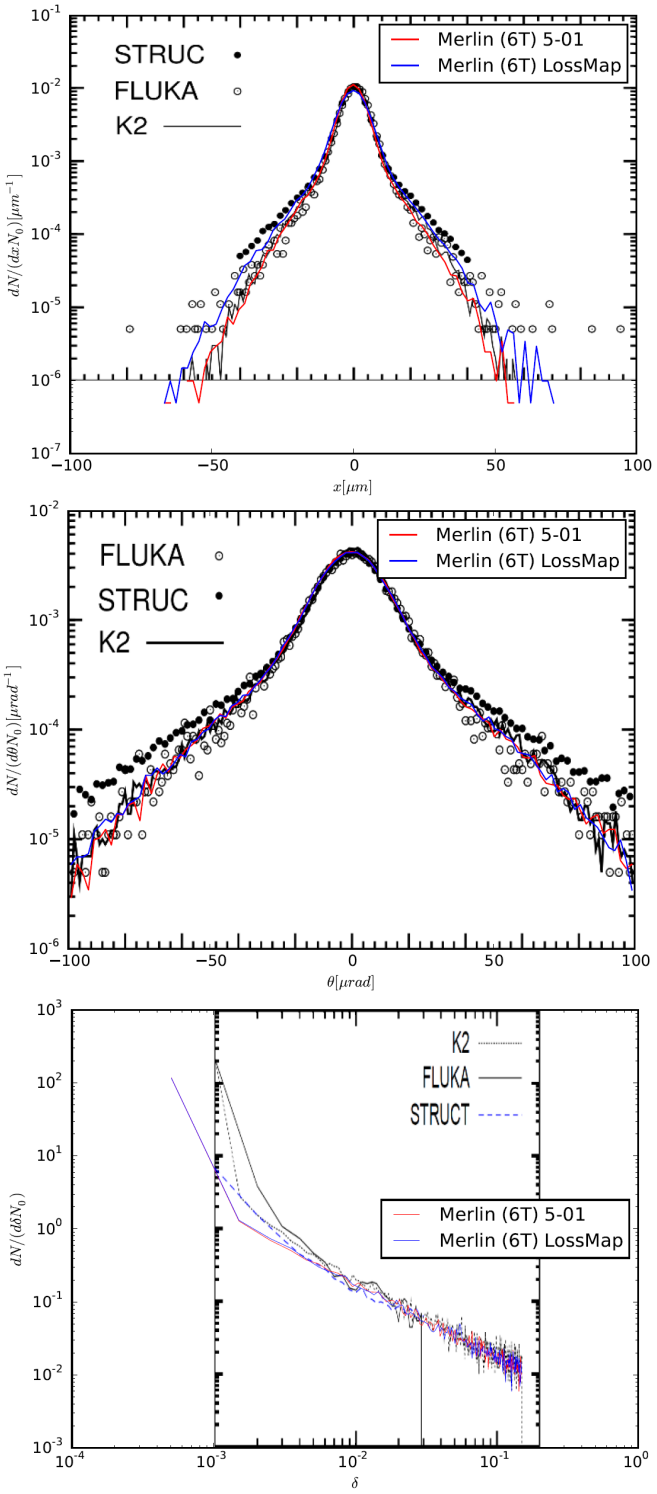

Figure 5.11: Scattering probabilities for a 7 TeV proton impacting on a 0.5 m long Cu jaw. Change in position (top), angle (middle) and energy (bottom) are compared for MERLIN 5-01 (red) and the loss map version of MERLIN (blue). Both cases use SixTrack+K2 like scattering. The data is overlaid onto that from the same study performed using other codes in [94]. In this case the δ distribution is in better agreement with K2, this is no surprise as the scattering processes used are based on K2.

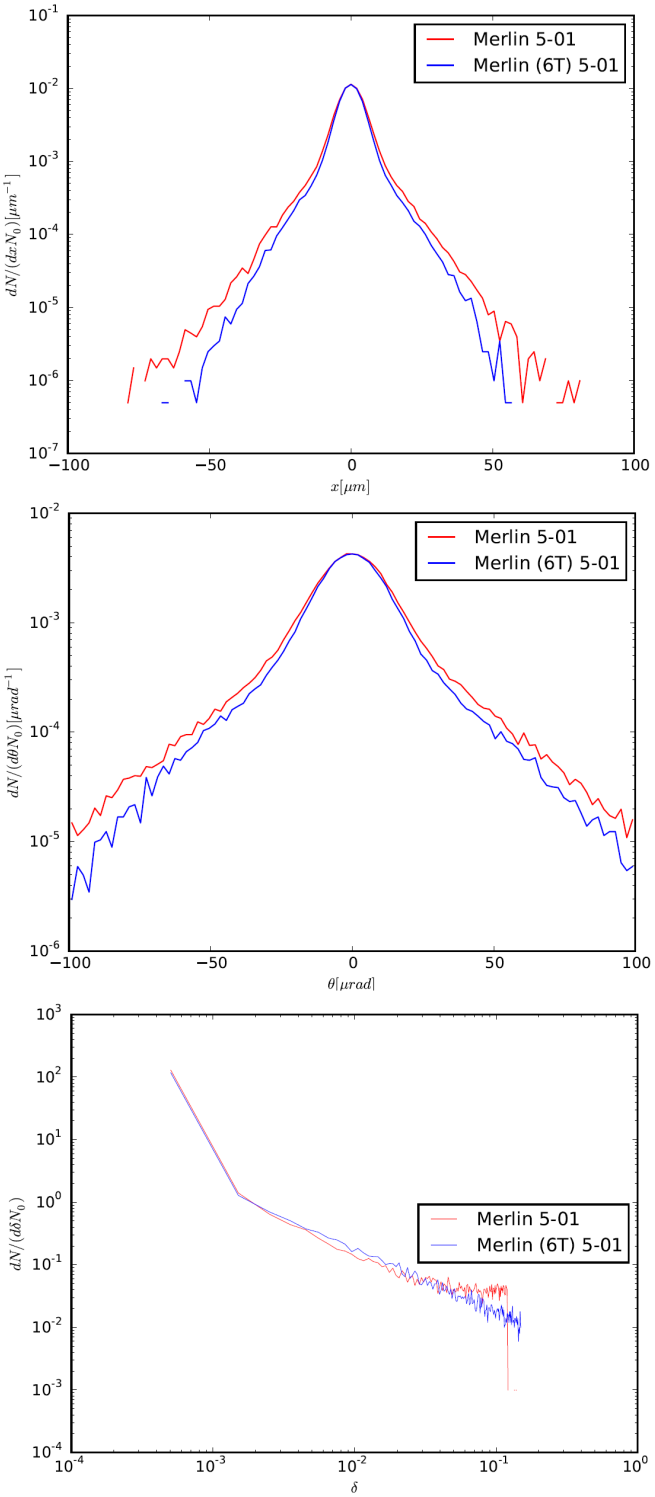

Figure 5.12: Scattering probabilities for a 7 TeV proton impacting on a 0.5 m long Cu jaw. Change in position (top), angle (middle) and energy (bottom) are compared for MERLIN (red) and SixTrack+K2 like scattering (blue). The difference in δ is is made clear from the bottom plot, with MERLIN scattering givint a lower cut-off.

5.4.2 Choice of Collimator Jaw Materials

The test case was repeated with different materials in order to compare the effect of pure materials with composites. Figure 5.13 compares the use of pure carbon with CFC AC150K, it is clear that though the distributions have similar ranges, more particles survive the interaction with AC150K. This is also the case when comparing pure copper with the physical counterpart - GlidCop, and pure tungsten with Inermet180, as shown in Fig. 5.14. As the collimators are made of these three materials, treating them as pure elements gives rise to an underestimation of the escaped halo and thus non-collimator losses.

To directly compare the use of the ST-like homogeneous composite with the MERLIN composite material, as well as the MERLIN and ST-like scattering routines, CuCD and MoGr are used in Fig. 5.15. In both figures, the difference between MERLIN and ST-like scattering is as noted in the previous section, with MERLIN scattering providing a slightly larger spread in the transverse position and angle, and a sharper cut-off in momentum. The effect of scattering is much more prevalent than the method of composite material treatment, with the difference between MERLIN and ST-like scattering much more obvious than that from using different composite methods. This provides a preliminary conclusion - that the homogeneous composite method used in SixTrack appears equivalent to MERLIN's more complex method - as at these length scales there is no discernible divergence. This implies that the more computationally expensive method of composite material implementation is not necessary.

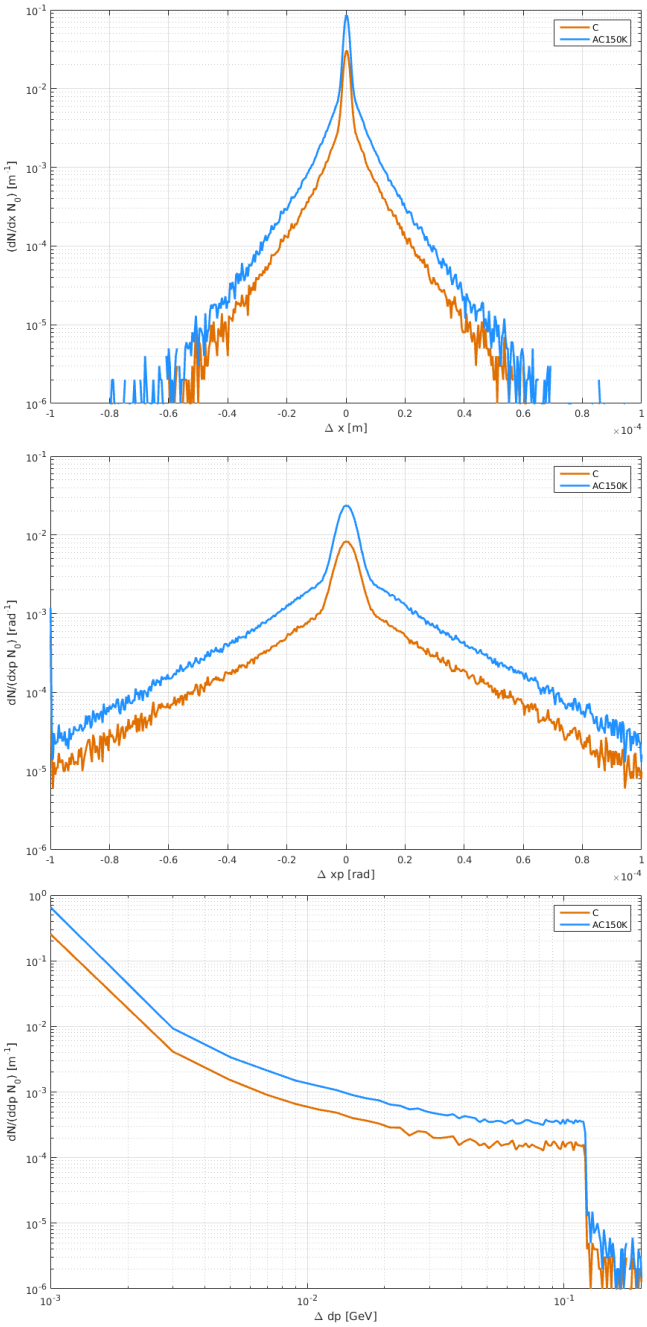

Figure 5.13: Scattering probabilities for a 7 TeV proton impacting on a 0.5 m long collimator jaw. Change in position (top), angle (middle) and energy (bottom) are compared for C and AC150K, note that neither of these materials are treated as composites in MERLIN. It is clear that the composite treatment is similar in behaviour to the pure element, however the frequency of scatters is much higher when using the composite.

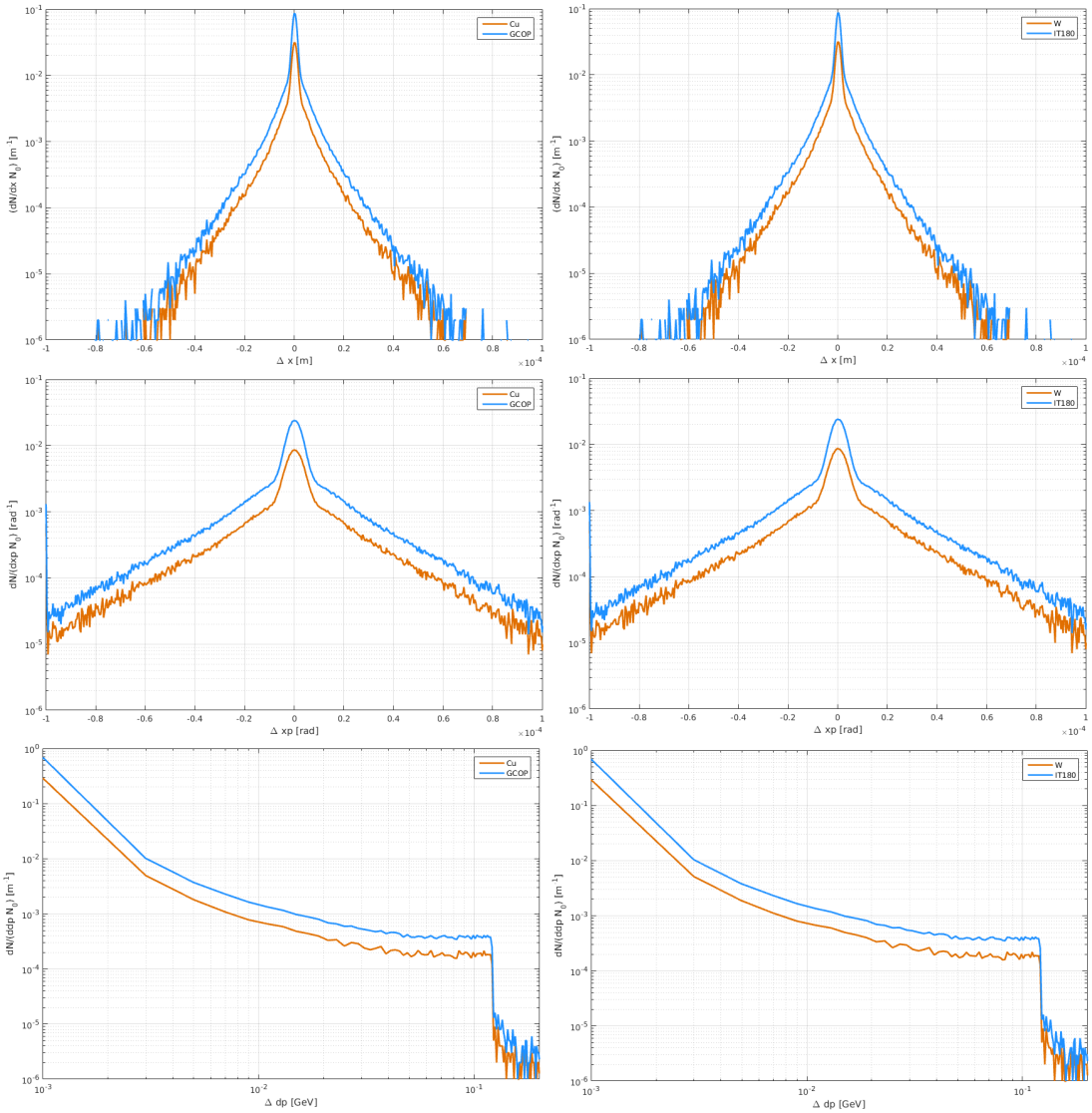

Figure 5.14: Scattering probabilities for a 7 TeV proton impacting on a 0.5 m long collimator jaw. Change in position (top), angle (middle) and energy (bottom) are compared for the pure material Cu and composite material GCOP (left), and for the pure material W and composite material IT180 (right). Again both comparisons show similar behaviour between pure elements and composites, with composites giving a larger number of scatters for all materials.

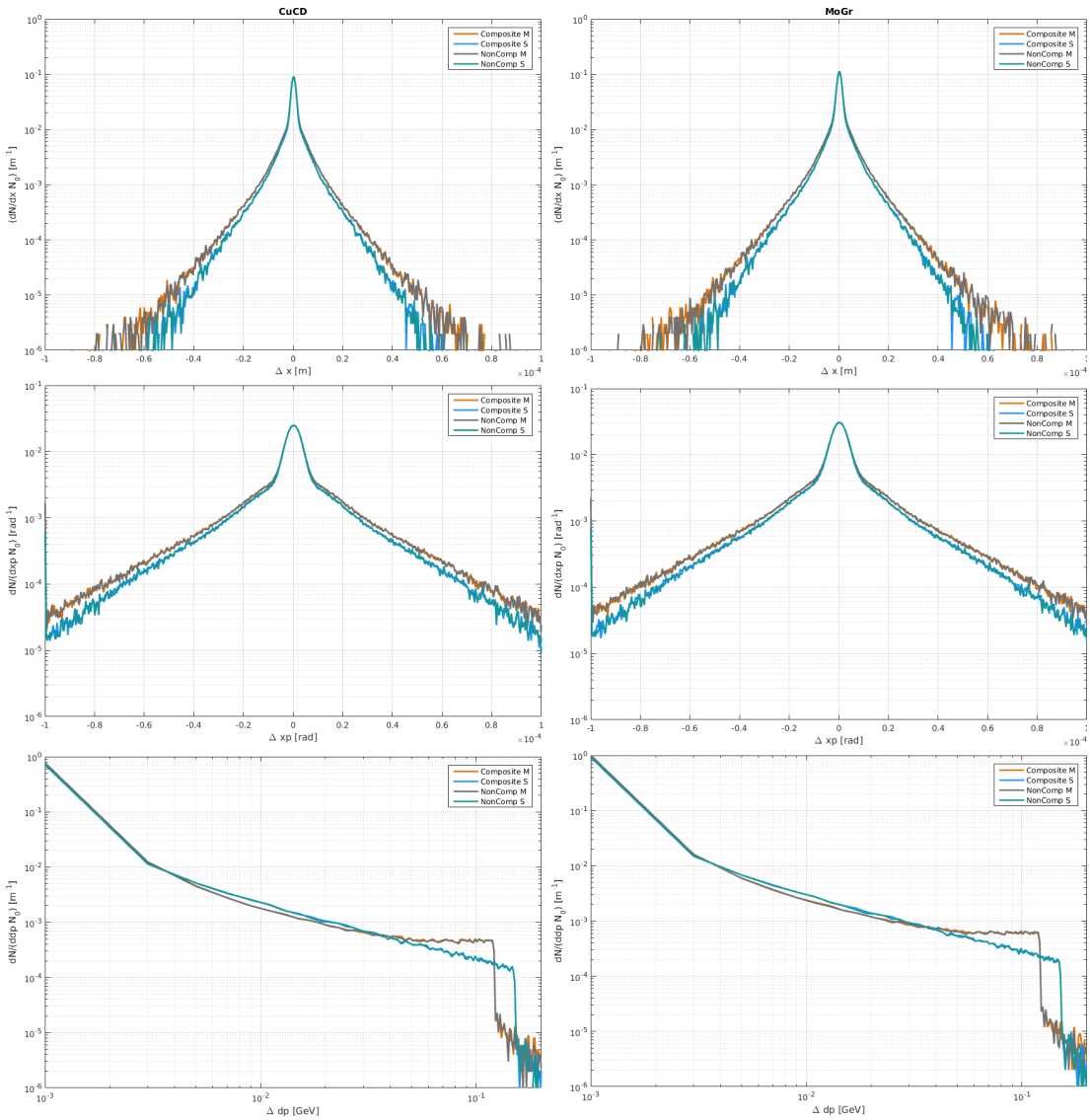

Figure 5.15: Scattering probabilities for a 7 TeV proton impacting on a 0.5 m long collimator jaw. Change in position (top), angle (middle) and energy (bottom) are compared for CuCD (left) and MoGr (right) as a composite and ST-like homogeneous composite, using MERLIN and ST-like scattering for each. For these materials the difference between treating them as a composite and pure material appears negligible. The difference between MERLIN and ST-like Scattering is clear however.

5.4.3 Scattering Processes

In this section we perform a comparison of individual scattering processes. The simulation set up is similar to that used in the previous test case, however we use a 1 cm long collimator with no jaw opening. Impacting a pencil beam of protons upon this solid block of material at the origin ($(x, y) = (0, 0)$) provides a scattering thickness that is less than the mean free path for protons in the tested materials, and thus it is more likely that only a single scattering process will take place. Using the tagging of scattering processes (the particle `type` co-ordinate), we may selectively output and histogram the effect of individual scattering processes over this thickness of material.

It must be noted that, whereas a thickness of 1 cm decreases the likelihood of multiple point-like scattering processes taking place, it does not guarantee that more scatters will not occur. Multiple Coulomb scattering (MCS) and energy loss due to ionisation will occur along the entire 1 cm length. In the following section the momentum transfer is calculated using the final particle momentum offset δ or dp, rather than the exact momentum change due to the scattering process, thus it is the sum of the scattering process and ionisation momentum transfers. The polar angle θ is also calculated from the final particle co-ordinate, and is thus the sum of MCS and any individual scattering processes. This approach offers a valid method of comparison as in most cases these quantities are small for MCS.

We observe from Fig. 5.16 that the effect of MCS and energy loss via ionisation produce a final particle polar angle of order 10^{-6} rad, and momentum transfer of order 10^{8} GeV ^{2}. These are both an order of magnitude smaller than any other scattering process and should not invalidate a direct comparison with SixTrack data, which is produced for the scattering process rather than over the 1 cm distance, demonstrated in the next section.

Figure 5.17 shows the distributions of particles that have undergone proton-nucleus elastic scattering. There is little difference between MERLIN and ST-like scattering with both giving an angular spread of around 10^{-5} rad, with the exception that the simpler ST-like routine generates a more uniform distribution in Δdp. This translates to a similar momentum transfer, with a range of around 10^{11} GeV ^{2}. Proton-nucleon elastic scattering is shown in Fig. 5.18, here there are two noteworthy differences; the uniform Δdp distribution generated in the ST-like scattering routine, as in proton-nuclear scattering, and the double peak at low $|x'|$ which translates into a peak at low θ. This double peak is currently not understood, as it is believed to be an artefact from the advanced proton-nucleon elastic scattering routine that is beyond the scope

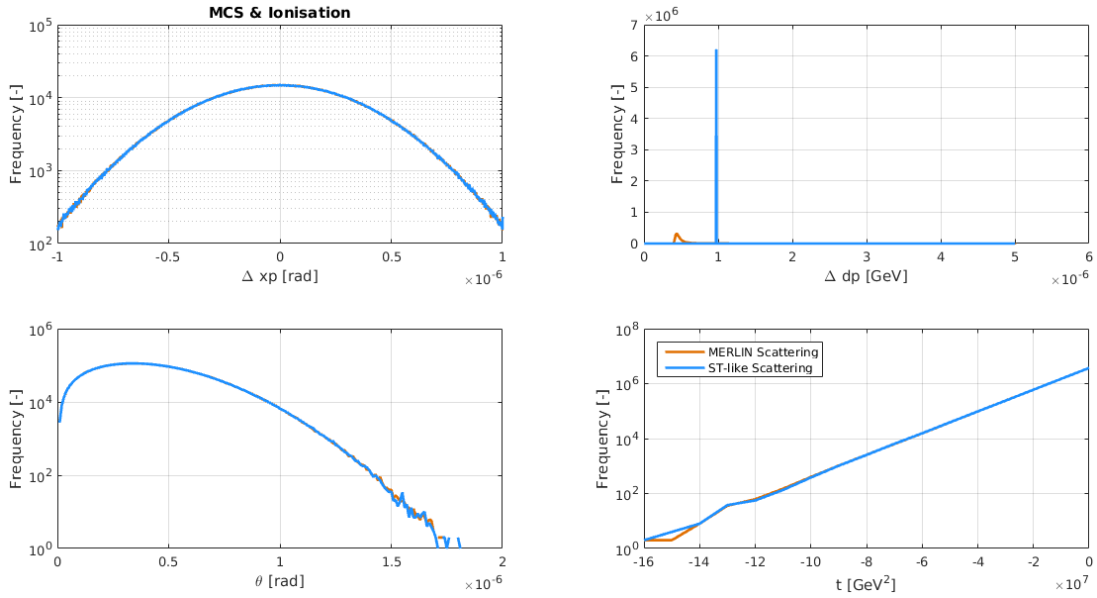

Figure 5.16: Co-ordinate distributions of $6.4 \cdot 10^6$ protons that have undergone multiple coulomb scattering and ionisation only, comparing MERLIN and ST-like scattering in MERLIN when traversing 1 cm of CFC AC150K.

of this thesis. This will be investigated in the future. The angular spread is an order of magnitude larger than the proton-nuclear elastic case at 10^{-4} rad, and the t range is increased by a factor ≈ 5. The proton-nuclear elastic interaction has a higher contribution to the total cross sections for protons at the LHC energy than the proton-nucleon, as shown in Table 5.3.

The largest momentum transfer is given by single diffractive interactions, as shown in Fig. 5.19. There is a clear difference between both routines in MERLIN, evidence of the improvements given by the advanced scattering routine [49] when compared to the ST-like. The range of 10^{12} GeV^2 is double that of proton-nucleon elastic, though the contribution to the total cross section is much smaller as shown in Table 5.3. This large Δdp range clarifies the origin of the difference between MERLIN and SixTrack like scattering in the 0.5 m collimator test case performed in section 5.4.1.

The number of point like scatters for $6.4 \cdot 10^6$ protons in 1 cm of each possible collimator jaw material is shown in Table 5.3. The percentage of interactions for each process is compared to the percentage of the process' cross section for non-composite elements (as composite elements have cross sections for each constituent). It is clear that in all cases the percentage of point-like scatters is equal to the cross section percentage,

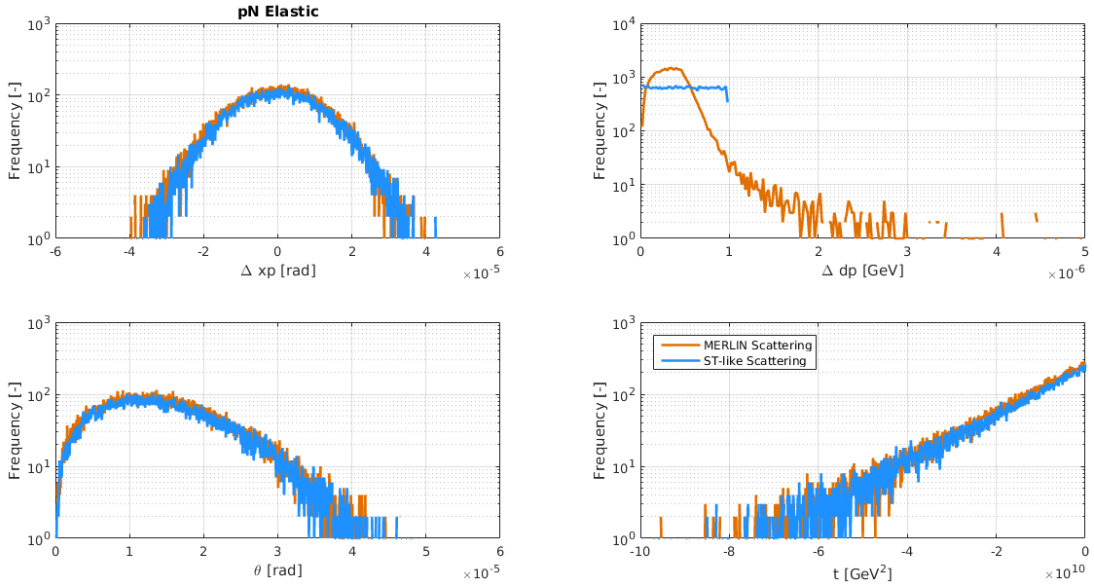

Figure 5.17: Co-ordinate distributions of $6.4 \cdot 10^6$ protons that have undergone MCS, ionisation, and proton-nucleus elastic scattering, comparing MERLIN and ST-like scattering in MERLIN when traversing 1 cm of CFC AC150K.

as expected.

It is also clear that, as shown in various figures in this section, the number of interactions is reduced when using the composite elements rather than their elemental counterparts. In the case of CFC AC150K this is due to different cross sections as this material is not treated as a composite in MERLIN. For GlidCop and Inermet this is due to the presence of elements that have lower interaction cross sections, which dilute the total cross section and thus the mean free path. In reality there is not a blanket reduction in the number of interactions, as this depends on the composite constituents.

The inelastic cross section includes cross sections for all interactions that produce secondary particles (with the exception of single diffractive dissociation), and contributes more than half of the total cross section for the materials detailed in Table 5.3, MERLIN treats an inelastic interaction as a lost proton.

Using the 1 cm collimator test case we have compared MERLIN and ST-like scattering within MERLIN, identifying that MCS and ionisation are identical but for the Δdp distributions, which though of different forms are similar in magnitude. The elastic scattering routines (proton-nuclear and proton-nucleon) are similar with the caveat that the ST-like routines generate a uniform Δdp distribution which translates to a similar momentum transfer. The advanced proton-nucleon elastic routine produces an

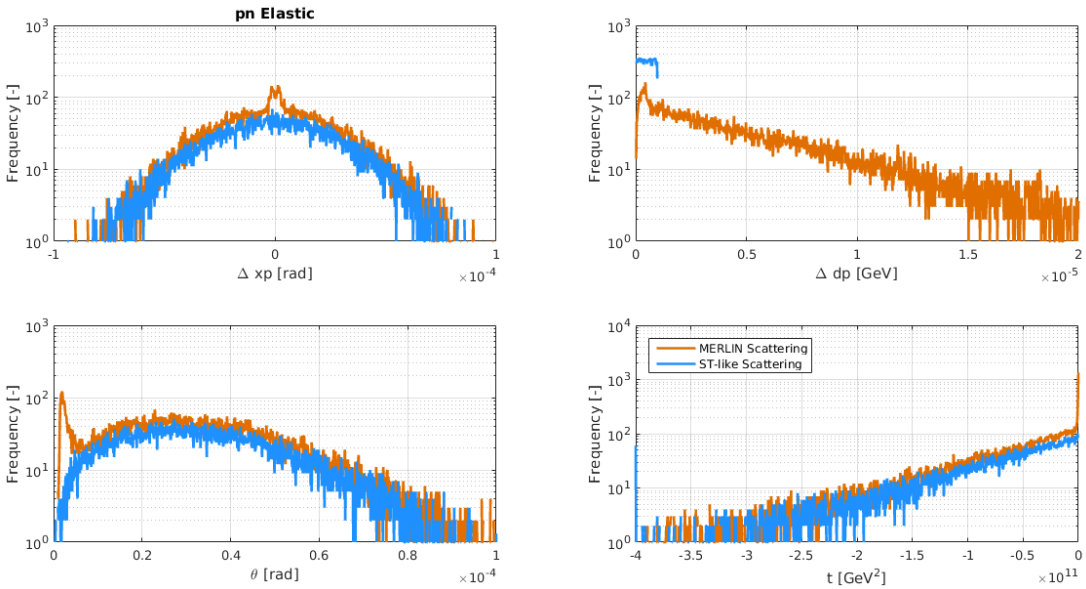

Figure 5.18: Co-ordinate distributions of $6.4 \cdot 10^6$ protons that have undergone MCS, ionisation, and proton-nucleon elastic scattering, comparing MERLIN and ST-like scattering in MERLIN when traversing 1 cm of CFC AC150K. The peak at low θ comes from the advanced proton-nucleon elastic scattering routine that is beyond the scope of this thesis, it is not understood and will be investigated in the future.

unexplained peak at low θ which requires further clarification. Single diffractive scattering is significantly different, as expected due to the work in [49]. This test case has allowed us to identify the difference in the previous 0.5 m collimator test *dp* distribution, which is due to the larger range given by the advanced single diffractive scattering routine in MERLIN.

5.4.4 Composite Material Implementation

The two main novel materials under study for LHC collimator jaw upgrades are molybdenum-carbide-graphite (MoGr), and copper-carbon-diamond (CuCD). A comparison of MCS, nucleon elastic, and single diffractive scattering between MERLIN and SixTrack is performed in this section for these materials. This includes the use of the MERLIN composite, the ST-like homogeneous composite in MERLIN, and data from a recent collimation version of SixTrack [93] [95], which uses the homogeneous composite method. We aim to identify any differences due to composite treatment, and advise as to the validity of the simple model as compared to the more advanced composite in MERLIN.

This study is interesting for two reasons, to identify differences between the compos-

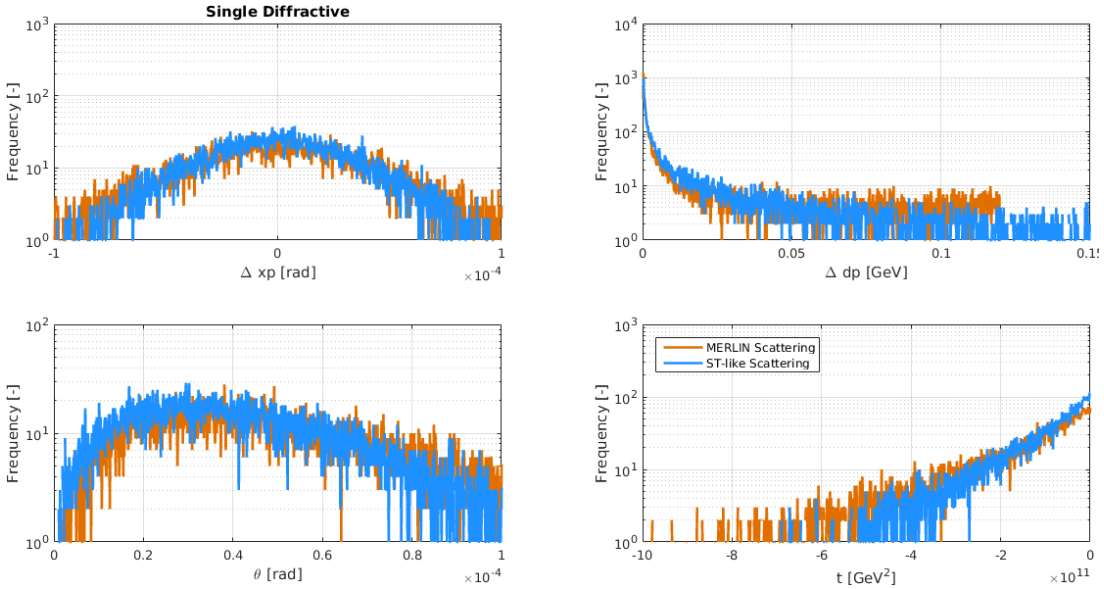

Figure 5.19: Co-ordinate distributions of $6.4 \cdot 10^6$ protons that have undergone MCS, ionisation, and single diffractive scattering, comparing MERLIN and ST-like scattering in MERLIN when traversing 1 cm of CFC AC150K. The Δdp cut-off when using MERLIN scattering explains the same cut-off shown in previous comparisons in Section 5.4.1, and is from the advanced single diffractive scattering routine which is described in [49].

ite material treatments, but also the difference in scattering routines - though MERLIN is equipped with ST-like scattering routines, these are not identical to the scattering in recent collimation versions of SixTrack. The SixTrack data is provided for the scattering interaction, *i.e.* the angular and momentum changes imparted to the proton only take into account the individual scatter, whereas in MERLIN these quantities are those of the particle after the scatter - where MCS and ionisation energy loss has taken place for any path length that the proton has traversed in the material. We have seen in the previous section that the contribution from MCS and ionisation energy loss is small compared to point-like scattering, and has negligible effect, however MCS and ionisation has been manually disabled in MERLIN where labelled in order to make a better comparison with SixTrack.

In the following figures we use **S** to denote SixTrack-like scattering in MERLIN, and **M** to denote MERLIN scattering. We also use **ST-like** to indicate the SixTrack-like homogeneous composite material method, and **Composite** for the more advanced MERLIN method. We plot histograms of frequency in order to compare the behaviour in each chosen co-ordinate, as well as the number of scattering events. We must also

	C	AC150K	W	IT180	Cu	GCOP	MoGr	CuCD
Inelastic								
Scatters	163478	123266	645388	618000	411811	410721	175929	311286
Scatter %	64.47	64.57	58.09	58.58	59.15	59.09	64.25	62.60
Cross Section %	64.47	-	58.05	-	59.07	-	-	-
pN Elastic								
Scatters	46378	34848	403812	372892	226426	225882	51559	113156
Scatter %	18.29	18.26	36.34	35.35	32.52	32.50	18.83	22.76
Cross Section %	18.39	-	36.32	-	32.61	-	-	-
pn Elastic								
Scatters	30789	22985	41537	43324	40178	40741	32563	51182
Scatter %	12.14	12.04	3.74	4.11	5.77	5.86	11.89	10.29
Cross Section %	12.06	-	3.78	-	5.78	-	-	-
Single Diffractive								
Scatters	12891	9474	17397	18055	17021	16897	13703	21427
Scatter %	5.08	5.11	1.57	1.71	2.44	2.43	5.00	4.31
Cross Section %	5.05	-	1.58	-	2.42	-	-	-
Rutherford								
Scatters	47	49	2962	2663	818	835	61	225
Scatter %	0.019	0.026	0.267	0.252	0.117	0.12	0.022	0.045
Cross Section %	0.021	-	0.27	-	0.117	-	-	-
Total								
Scatters	253583	190895	1111096	1054934	696254	695076	273815	497276

Table 5.3: Number and percentage of point like interactions for $6.4 \cdot 10^6$ protons impacting upon a 1 cm long solid material.

ascertain if the ST-like composite in MERLIN is similar to that in SixTrack in order to gauge if other differences between the codes may be ignored.

First we compare the effect of composite implementation on MCS and ionisation. We expect that these processes will be identical regardless of the methods used, as MCS and ionisation routines are similar between the codes. Our expectation is confirmed for MoGr in Fig. 5.20, in which all scattering and composite material methods show similar behaviour. CuCD is similar and thus not shown.

In the following figures the MCS and ionisation routines in MERLIN were switched off manually in order to compare with the SixTrack data, the figures are labelled as such. The spread in the transverse horizontal momentum (or angle) component x' due to nucleon elastic scattering is shown in Fig. 5.21 for CuCD. The agreement between SixTrack and the ST-like composite is clear, indicating that the models are similar. The difference between the two scattering methods in MERLIN is negligible in this co-ordinate, however there is clear disparity between the homogeneous composite and the MERLIN composite, which has a larger spread in the tails at larger x'.

The nucleon elastic x' is shown in Fig. 5.22 for MoGr, it is interesting to note that for this material there is no noticeable difference between scattering or composite implementation. This is likely due to MoGr being mostly (97.3%) carbon, whereas CuCD contains 25.7% copper and 73.3% carbon diamond. The homogeneous mixture of MoGr is dominated by carbon and will behave similarly to the composite material, which is not the case for CuCD.

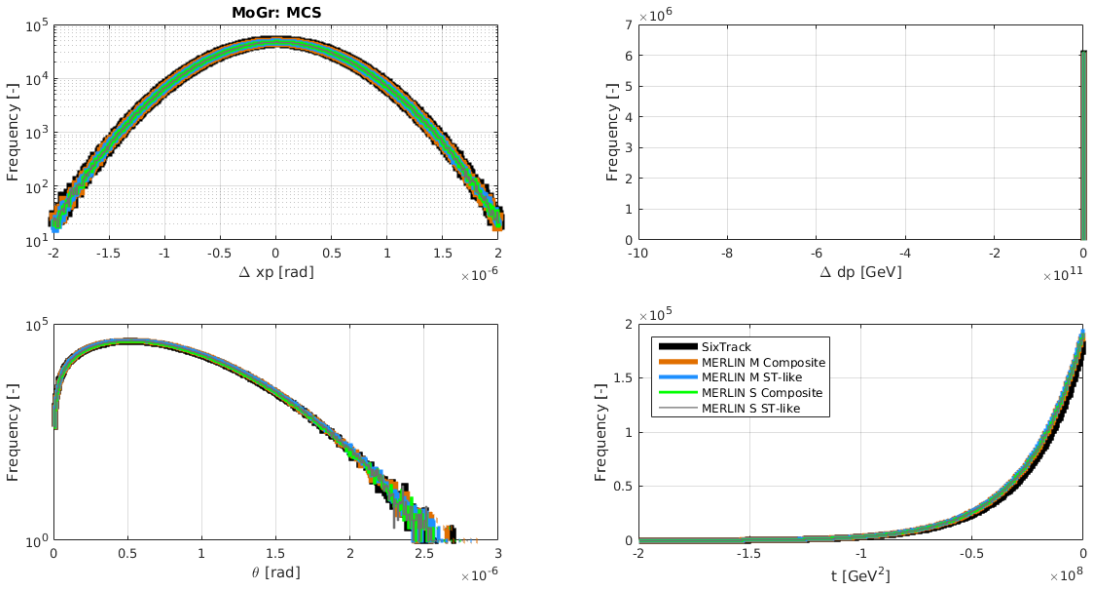

Figure 5.20: Co-ordinate distributions of $6.4 \cdot 10^6$ protons that have undergone multiple coulomb scattering and ionisation only, when traversing 1 cm of MoGr. Line thickness is exaggerated in order to clarify that, as expected, the data is similar for all cases.

The magnitude of the change in momentum deviation co-ordinate δ (or dp) for single diffractive scattering is compared in Fig. 5.23. We observe little difference between the codes when using ST-like scattering, the SixTrack data showing a small increase in frequency of scatters at dp < 0.05 GeV. There is a significant change in the Δdp cut-off when using MERLIN's more advanced single diffractive scattering. This shows that the ST-like implementation of SD scattering in MERLIN is in good agreement with SixTrack, and also that MERLIN's more advanced SD routine provides a larger frequency of high momentum transfer scatters, though the ST-like and SixTrack scattering give a larger overall range of momentum transfer. The cause of this difference is beyond the scope of this thesis as it has been covered in detail in [49]. MoGr is not shown as the results are similar to CuCD for this case.

When looking at the polar angle distribution of single diffractive scattering in Fig. 5.24, it appears that when using ST-like scattering SixTrack again generates more interactions than MERLIN but otherwise the behaviour is similar. For MERLIN scattering there is clearly a difference with SixTrack, though again the effect of the composite implementation is negligible. The polar angle for this plot is shown for MoGr only, as CuCD is similar.

Figures 5.25 and 5.26 show the momentum transfer distributions for single diffrac-

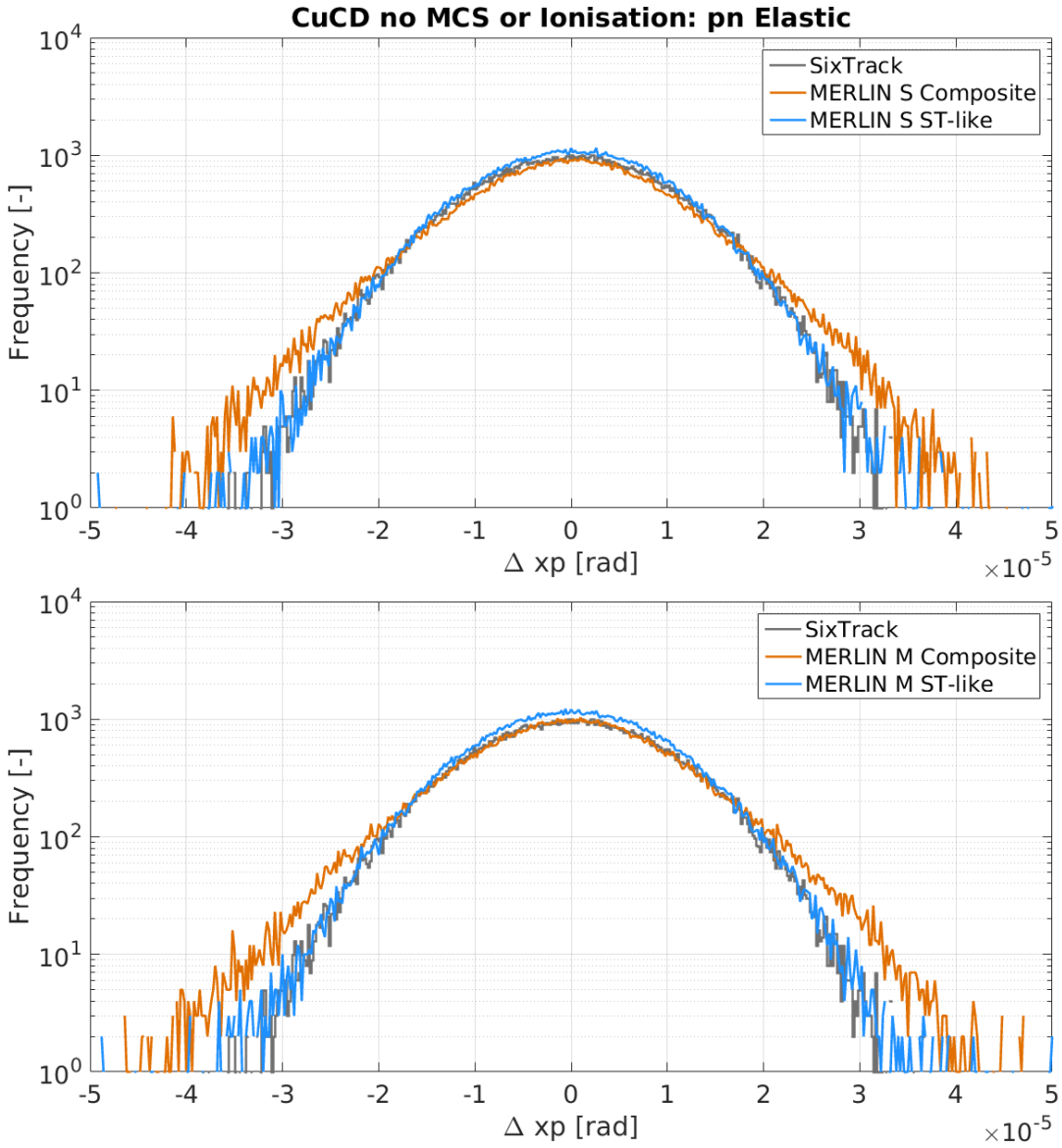

Figure 5.21: Transverse horizontal angle x' distribution of $6.4 \cdot 10^6$ protons that have undergone proton-nucleon elastic scattering when traversing 1 cm of CuCD. SixTrack-like scattering in MERLIN is shown above, with MERLIN scattering below. Treating CuCD as a composite gives a larger spread in angle. This shows that composite treatment can offer significant differences in scattering, and cannot be ignored despite the larger scale results in Section 5.4.1 (which appears to show no influence from modelling CuCD as a composite rather than a pure material).

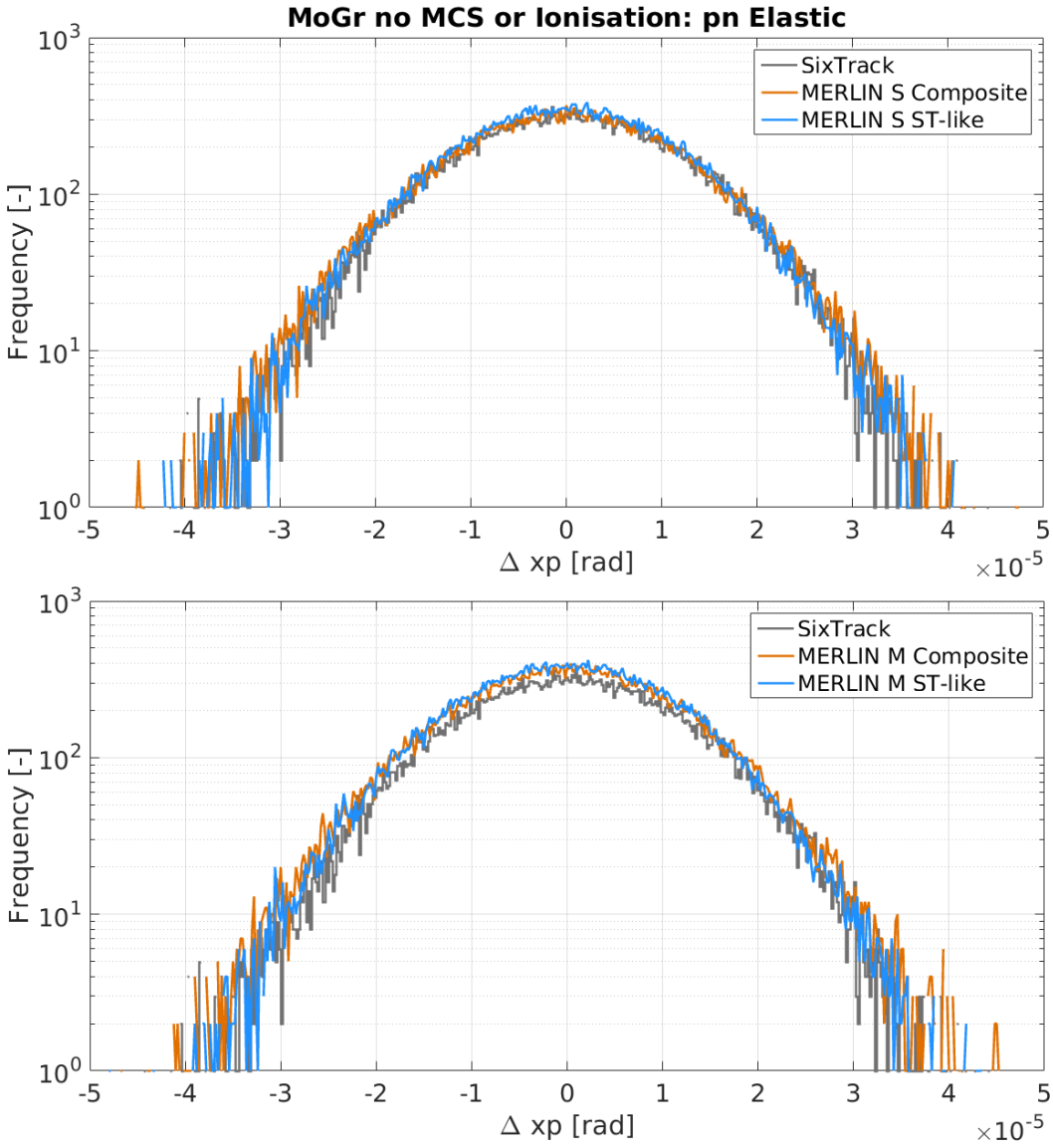

Figure 5.22: Transverse horizontal angle x' distribution of $6.4 \cdot 10^6$ protons that have undergone proton-nucleon elastic scattering when traversing 1 cm of MoGr. SixTrack-like scattering in MERLIN is shown above, with MERLIN scattering below. The difference observed in Fig. 5.21 is not observed here, this is due to MoGr being dominated by a single element (97.3% carbon).

tive scattering in CuCD and MoGr respectively. With ST-like scattering it is clear that there are a larger number of scatters for the MERLIN composite. With MERLIN scattering this is not so apparent, however there is a larger spread, and larger number

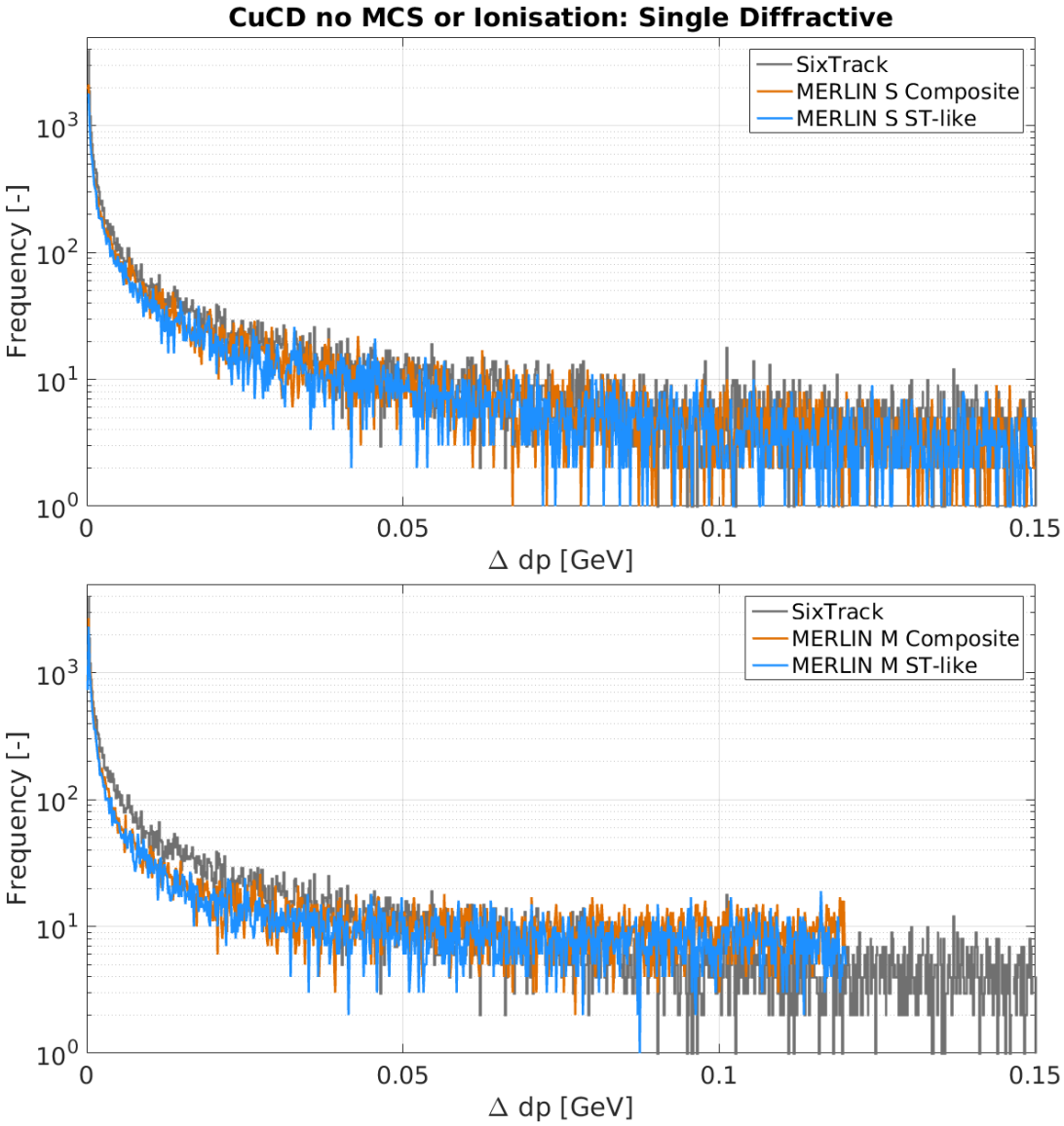

Figure 5.23: Momentum offset magnitude $|\delta|$ distribution of $6.4 \cdot 10^6$ protons that have undergone single diffractive scattering when traversing 1 cm of CuCD. SixTrack-like scattering in MERLIN is shown above, with MERLIN scattering below. The momentum offset cut-off in the bottom plot is the hallmark of the advanced single diffractive scattering in MERLIN.

of scatters when compared to the homogeneous composite in SixTrack. This is of particular importance when considering losses in areas of high dispersion. The momentum transfer for single diffractive scattering has the largest range, this will produce the ma-

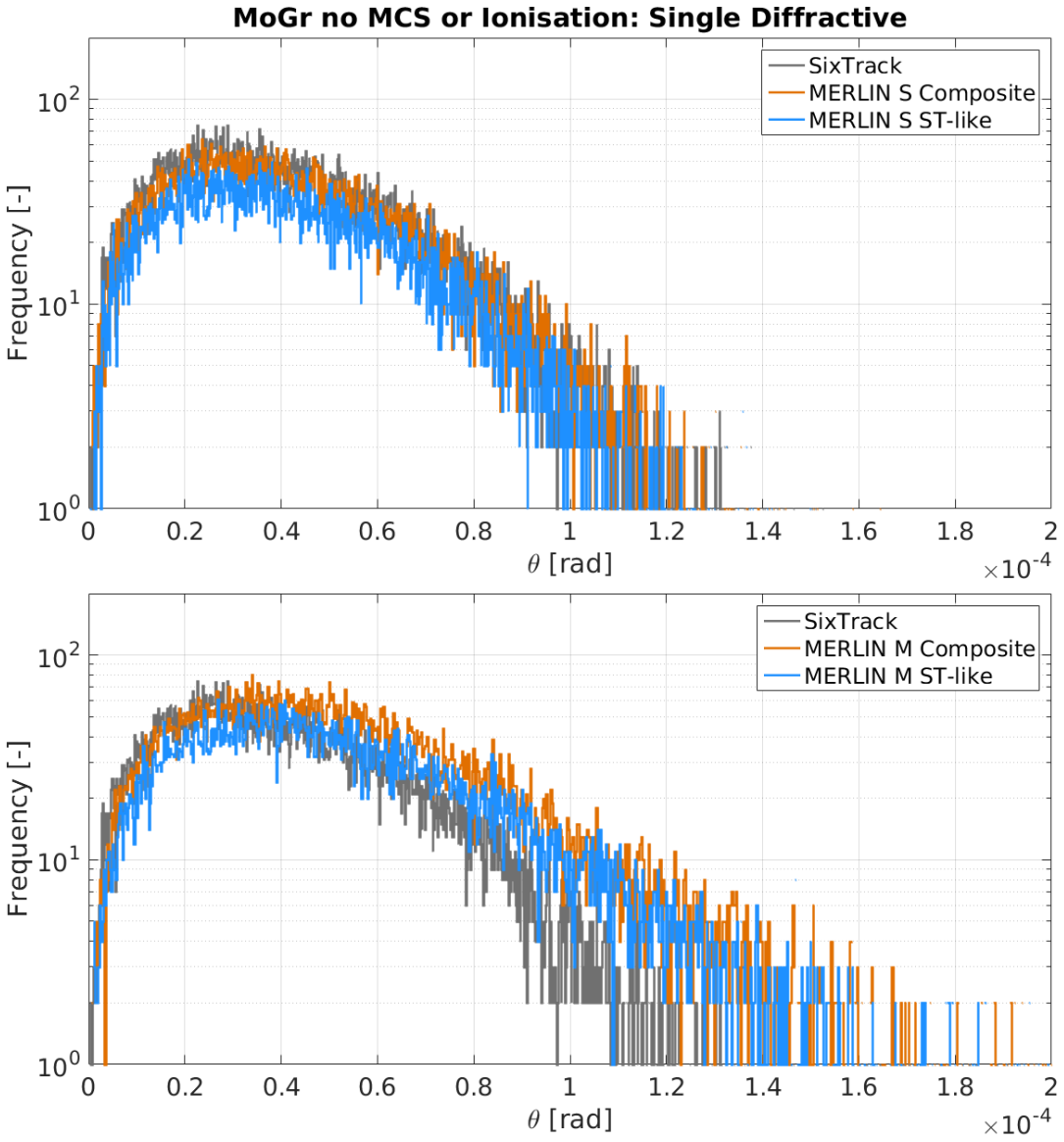

Figure 5.24: Polar angle θ distribution of $6.4 \cdot 10^6$ protons that have undergone proton nucleon elastic scattering when traversing 1 cm of MoGr. SixTrack-like scattering in MERLIN is shown above, with MERLIN scattering below. The advanced single diffractive scattering in MERLIN gives a larger θ range than that in SixTrack as shown in the bottom plot.

jority of secondary halo protons that are lost in the dispersion suppressor regions post collimation in IR7.

Table 5.4 compares the percentage of total scatters in MERLIN and ST-like ho-

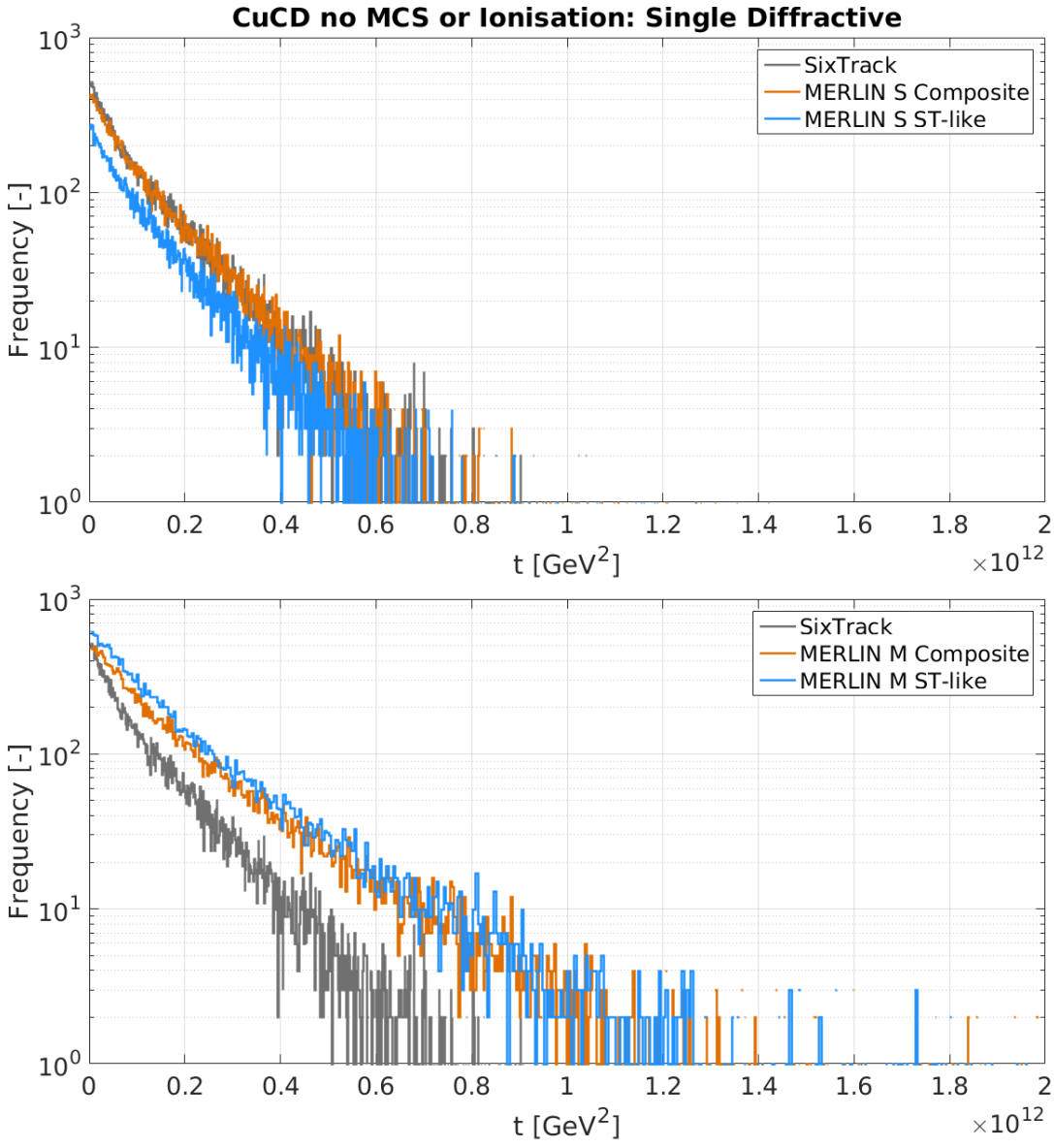

Figure 5.25: Momentum transfer magnitude $|t|$ distribution of $6.4 \cdot 10^6$ protons that have undergone proton nucleon elastic scattering when traversing 1 cm of CuCD. SixTrack-like scattering in MERLIN is shown above, with MERLIN scattering below.

mogeneous composites attributed to each scattering process. It is evident that the homogeneous composite cross sections do not reproduce an accurate representation of the expected scattering types, though the total number of scatters may be similar. This is particularly clear for CuCD and MoGr, where some scatters differ by more than a

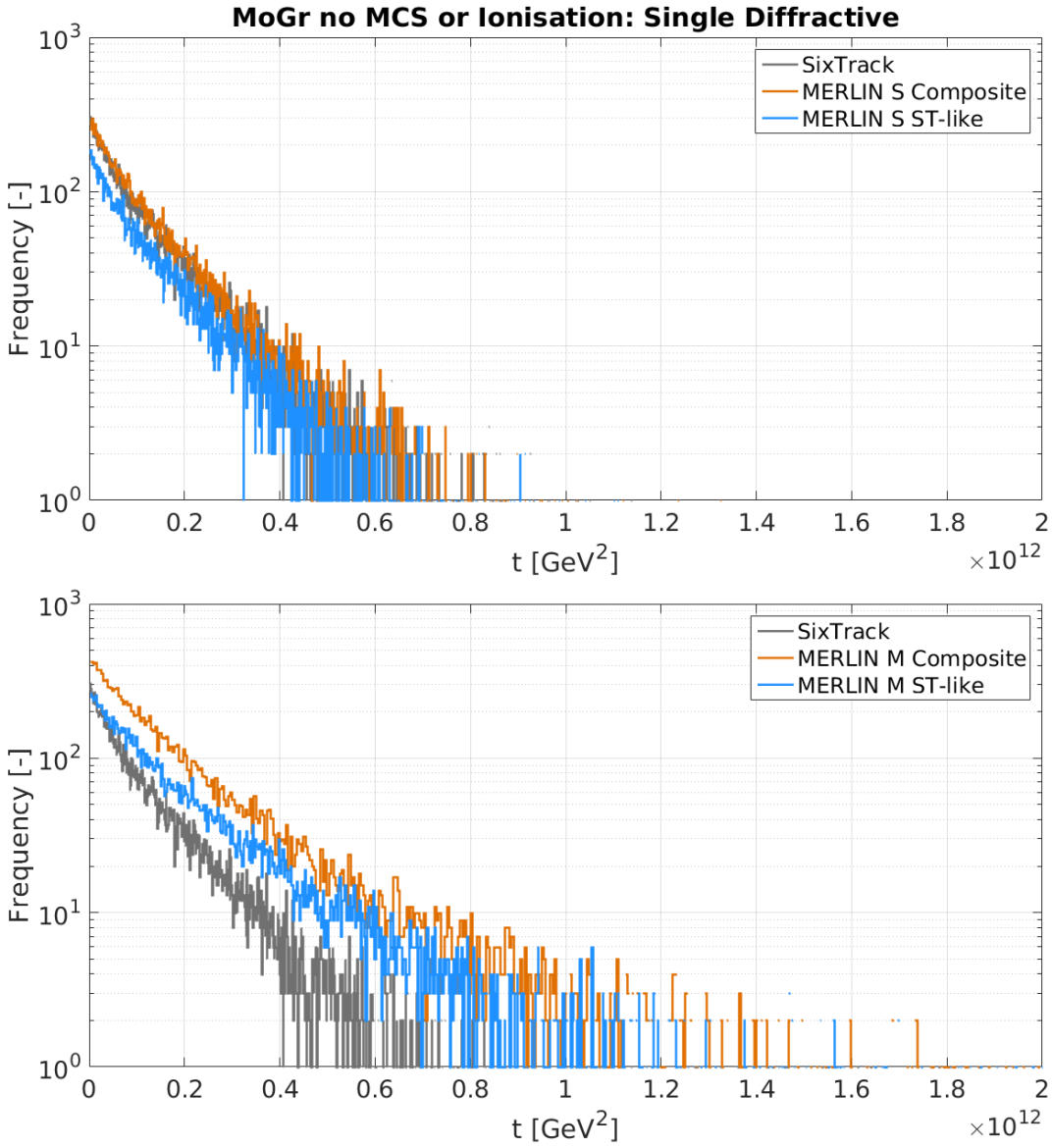

Figure 5.26: Momentum transfer magnitude $|t|$ distribution of $6.4 \cdot 10^6$ protons that have undergone single diffractive scattering when traversing 1 cm of MoGr. SixTrack-like scattering in MERLIN is shown above, with MERLIN scattering below.

per cent.

	IT180	GCOP	MoGr	CuCD
Inelastic				
ST-like %	58.29	59.06	63.76	60.84
MERLIN %	58.58	59.09	64.25	62.60
pN Elastic				
ST-like %	35.82	32.58	19.54	26.12
MERLIN %	35.35	32.50	18.83	22.76
pn Elastic				
ST-like %	3.99	5.79	11.81	9.08
MERLIN %	4.11	5.86	11.89	10.29
Single Diffractive				
ST-like %	1.65	2.45	4.86	3.88
MERLIN %	1.71	2.43	5.00	4.31
Rutherford				
ST-like %	0.257	0.12	0.029	0.074
MERLIN %	0.252	0.12	0.022	0.045
Total				
ST-like Scatters	1055230	695259	274324	497225
MERLIN Scatters	1054934	695076	273815	497276

Table 5.4: Percentage of point like interactions for $6.4 \cdot 10^6$ protons impacting upon a 1 cm long solid material, comparing the ST-like homogeneous composite with MERLIN's more advanced composite.

When comparing SixTrack with ST-like scattering and composite treatment in MERLIN (labelled MERLIN S ST-like), we observe that the number of single diffractive scatters is always higher in SixTrack. In this case we are using imitations of SixTrack's scattering and homogeneous composite method, the fact that MERLIN's ST-like data is not identical to SixTrack shows that there is a small difference between the codes. As the scattering in SixTrack has recently been updated [91] this is expected. Taking this into account, the number of single diffractive interactions is increased when using the MERLIN composite as compared to the homogeneous composite.

We now have a number of tentative conclusions on the effect of the two methods of implementing composite materials, and the use of ST-like or MERLIN scattering routines, for the 1 cm test case.

- As expected there is negligible difference in MCS and ionisation between both codes and all methods of composite or scattering implementation.
- The new single diffractive scattering routine in MERLIN offers different behaviour from that in SixTrack, this was previously observed on the scale of a loss map

in [49] but has been clarified at a much smaller scale here.

- The effect of different composite material implementations is more pronounced for CuCD. Though it was not previously noticeable in the 0.5 m test case, it has been shown to be important on a small scale, due to the material composition. This implies that when a composite material is not dominated by a single element (as in the case of CuCD), the homogeneous composite is not a valid approximation. In the case of proton-nuclear elastic scattering in CuCD it is clear that the composite treatment has more impact than the choice of scattering routine.
- In some instances it is clear that the scattering routine has a larger impact than the composite treatment, for example single diffractive scattering. This means that both are important considerations in successful modelling of proton interaction with LHC collimators.
- When using a combination of ST-like scattering and homogeneous composite method in MERLIN, the results are similar to that of SixTrack, with a difference in frequency of scatters only, likely due to differing cross sections arising from recent updates to SixTrack [91].

5.4.5 Summary

We have compared MERLIN 5-01 with an existing set of results from [94], and have obtained an agreement better than previous versions of MERLIN. The slight difference in dp is explained by the use of a new and improved single diffractive scattering routine [96]. A more rigorous comparison with codes such as FLUKA may be done, this was started in [49] however only for the loss map version of the code which has since been merged and updated, this is envisioned for MERLIN 5.01 in the future.

Using the set up of this test case we have compared scattering in pure elements with the corresponding composite materials used for LHC collimators, in all cases investigated the number of interactions was higher when using the composite. This is important for simulating the interaction between halo protons and the LHC collimators as it will affect the number of particles that form the secondary halo, as well as the condition of the particles. When comparing MERLIN and ST-like scattering in the 50 cm test case the difference was observed to be negligible on this length scale.

The second test case of a pencil beam impacting upon a 1 cm long solid material was used to show that MERLIN's imitations of SixTrack (scattering routine and homogeneous composite) were in good agreement with SixTrack data. The effects of each scattering process were compared, and the advanced single diffractive routine differs significantly from previous models. An unexpected peak in the θ distribution given by the advanced proton nucleon elastic scattering was observed, and must be clarified in the future. MERLIN scattering offers differences which have already been identified as improvements when compared to experimental data [49], here we simply quantify the differences.

Homogeneous composites were compared with the more advanced MERLIN method, from which it was evident that when a composite is not dominated by a single element the homogeneous approximation is not valid. Thus MERLIN offers a more robust method of treating composite materials. This work is ongoing [97], and a more detailed comparison is expected in the future.

In conclusion we have validated MERLIN scattering and composite treatment against a number of well established codes, and where they are not in agreement it has been shown that MERLIN offers an improved method of modelling. We have also identified that the treatment of composite materials is important, and that the homogeneous method used in SixTrack has been improved upon.

5.5 Collimation

To validate MERLIN as a collimation tool for the LHC, we use the 6.5 TeV beam 2 lattice at flat top to compare MERLIN loss map simulation results with simulated SixTrack and measured loss maps. The beta functions, dispersion, aperture, and closed orbit have been compared for this case at the start of the chapter, and were found to be identical for MERLIN and SixTrack inputs. Table 5.5 shows the collimator settings used for the 6.5 TeV collimation simulations.

Insertion Region	**Collimator Family**	**Setting** [σ]
7	Primary	5.5
	Secondary	8
	Absorber	14
3	Primary	15
	Secondary	18
	Absorber	20
1	Tertiary	37
	Absorber	999
5	Tertiary	37
	Absorber	15
6	Dump Protection	9.1
	Secondary	9.1
2	Tertiary	37
	Injection	999
8	Tertiary	37

Table 5.5: LHC collimator settings used for the nominal 6.5 TeV beam 2 simulations in MERLIN, SixTrack, and measurement.

Figures 5.27 to 5.30 compare the initial distribution used in MERLIN with that used in the SixTrack simulations. In both cases a distribution between 5.5-5.54 σ_x is used in the transverse horizontal plane, and a 2 σ_y Gaussian distribution is used in the transverse vertical plane. We observe that the initial bunch distributions are slightly dissimilar between the codes, most noticeably the highest concentration of particles in transverse real space is more compact in MERLIN. This is simply due to a difference in the definition of the initial halo, and may give rise to nonconformity in loss map simulations. The majority of particles exist at around 5.5 σ, whereas in SixTrack the particles are initially between 4 - 5.54 σ, thus more particles will impact upon the primary collimator in early turns in MERLIN. Normally loss maps are simulated using a purely transverse distribution, in which case the effect of 6D tracking is difficult to compare to that of 4D, thus we use the 2 σ_z Gaussian distribution in ct dp longitudinal

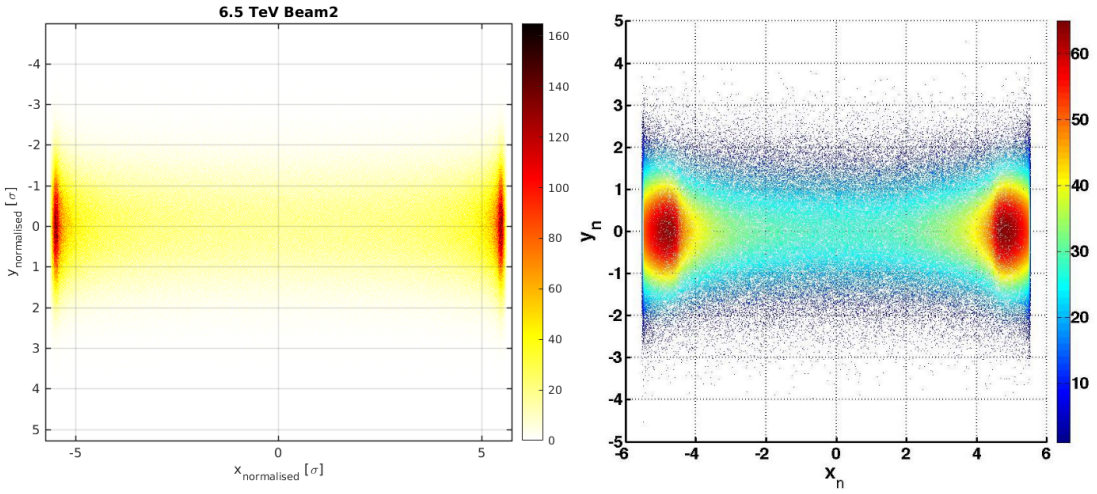

Figure 5.27: Initial distribution in real space used for 6.5 TeV beam 2 loss maps, comparing that used in MERLIN (left) with SixTrack (right). We observe that there is some difference in the way that MERLIN and SixTrack generate an initial halo bunch, with SixTrack giving a larger transverse spread.

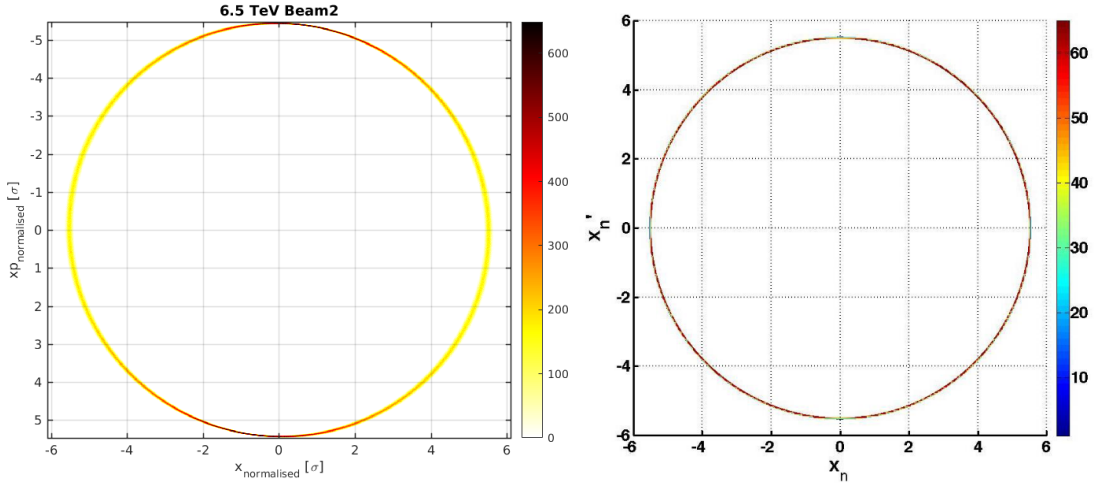

Figure 5.28: Initial distribution in xx' phase space used for 6.5 TeV beam 2 loss maps, comparing that used in MERLIN (left) with SixTrack (right). In this phase space the distributions are similar.

phase space.

We begin by ascertaining the effect of using `SYMPLECTIC` and `TRANSPORT` tracking and the importance of 6D tracking is clarified, followed by a brief comparison of simulating collimator jaws as composite materials rather than pure elements. Loss maps

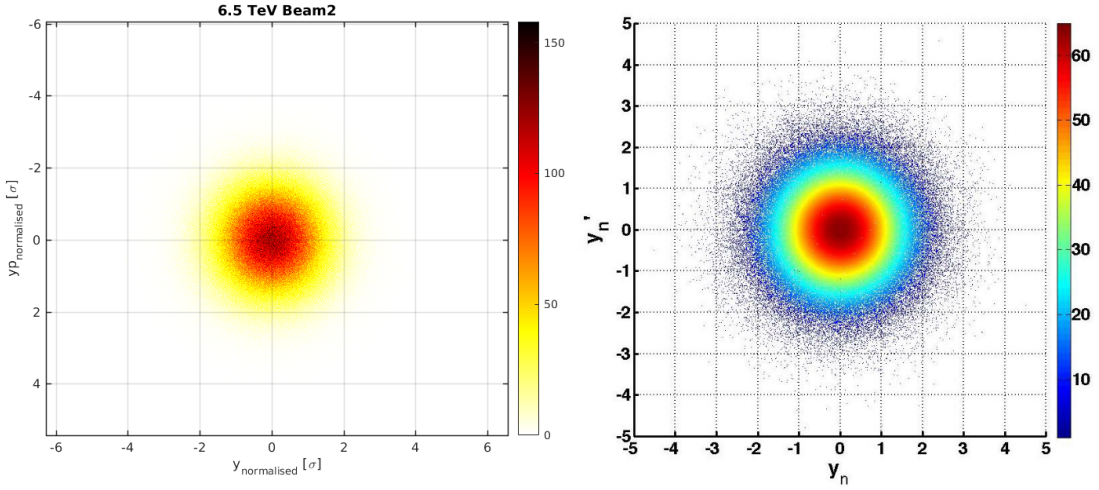

Figure 5.29: Initial distribution in yy' phase space used for 6.5 TeV beam 2 loss maps, comparing that used in MERLIN (left) with SixTrack (right). In this phase space the distributions are similar.

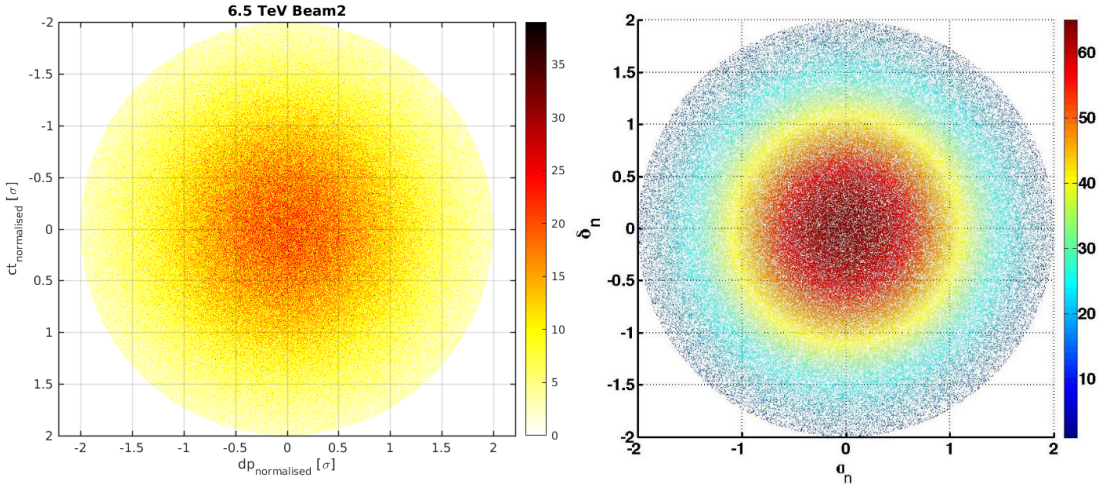

Figure 5.30: Initial distribution in longitudinal *ctdp* phase space used for 6.5 TeV beam 2 loss maps, comparing that used in MERLIN (left) with SixTrack (right). In this phase space the distributions are similar.

are plotted for the entire length of the lattice, for the LHC we use beam 1 co-ordinates, where $s = 0$ is defined at IP1.

5.5.1 Effect of Integrators and 6D Tracking

The difference between 4D and 6D tracking is shown in Fig. 5.31, this is clarified in Fig. 5.32 - without 6D tracking there are orders of magnitude fewer losses in IR3 (s = 6000 - 7000 m). IR7 is the focus of our concern for the largest collimation losses, Fig. 5.33 appears to show that we gain little from using 6D tracking, however with the increased number of lost particles in the momentum collimation insertion, the number of losses in IR7 decreases.

Table 5.6 compares the number of losses for each loss map simulation in this section. When using 6D tracking with a bunch that has a longitudinal component (momentum deviation or offset in ct) we observe that the total number of losses is larger than in the 4D case. Around $5 \cdot 10^5$ particles are lost in the momentum collimation region, reducing the losses in the betatron collimation region. With 4D tracking, there are almost no losses in the momentum collimation insertion.

There is a difference of around 50% in the IR3 collimator losses when comparing the trackers, this is likely due in part to the different RF cavity integrators as shown previously. Apart from this difference in the momentum collimation insertion, 6D tracking provides a more realistic prediction of loss locations for halo particles that lose momentum in collimators but re-enter the beam (*i.e.* the secondary and tertiary halos). 6D `SYMPLECTIC` tracking will be used for all following simulations where not otherwise stated.

Region	**Type**	**Losses**			
		4D Pure `SYMPLECTIC`	**6D Pure** `SYMPLECTIC`	**6D Composite** `TRANSPORT`	**6D Composite** `SYMPLECTIC`
Entire LHC	**Collimator**	6298362	6391014	6385850	6388154
	Cold	3010	3553	5356	4860
	Warm	240	425	726	663
IR7	**Collimator**	6294407	5819657	5503059	5771352
	Cold	2933	2463	3332	3230
	Warm	240	202	261	350
IR3	**Collimator**	2824	570358	880889	614758
	Cold	4	1030	1929	1537
	Warm	0	223	465	313

Table 5.6: Breakdown of losses in 4D, 6D,`TRANSPORT` and `SYMPLECTIC` loss maps for beam 2 of the 6.5 TeV LHC. Note that the 4D and 6D loss map simulations use pure materials, whereas the integrator comparison simulations use composite materials.

Figure 5.34 compares the use of the `TRANSPORT` and `SYMPLECTIC` integrator sets,

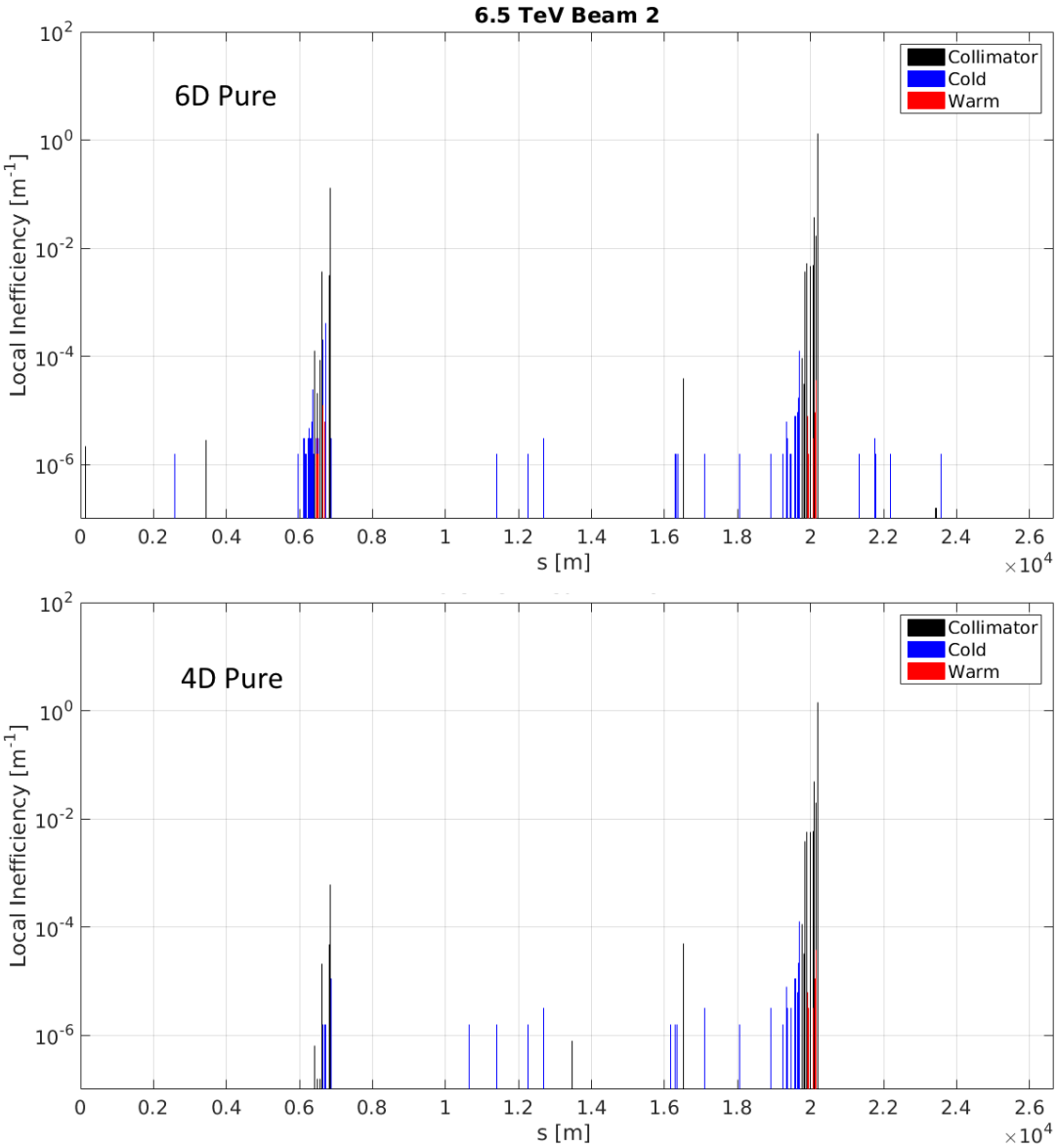

Figure 5.31: Loss maps for beam 2 of the 6.5 TeV LHC, plotted in the s co-ordinates for beam 1, using pure collimator materials and comparing the use of 6D tracking (above) with the use of 4D tracking (below). We observe that when using 6D tracking the losses in IR3 are significantly higher.

both using 6D tracking. The difference between these methods is negligible as expected, due to the fact that a loss map simulation requires only 200 turns, and thus does not present enough iteration for the `TRANSPORT` tracker to deviate from the Hamiltonian

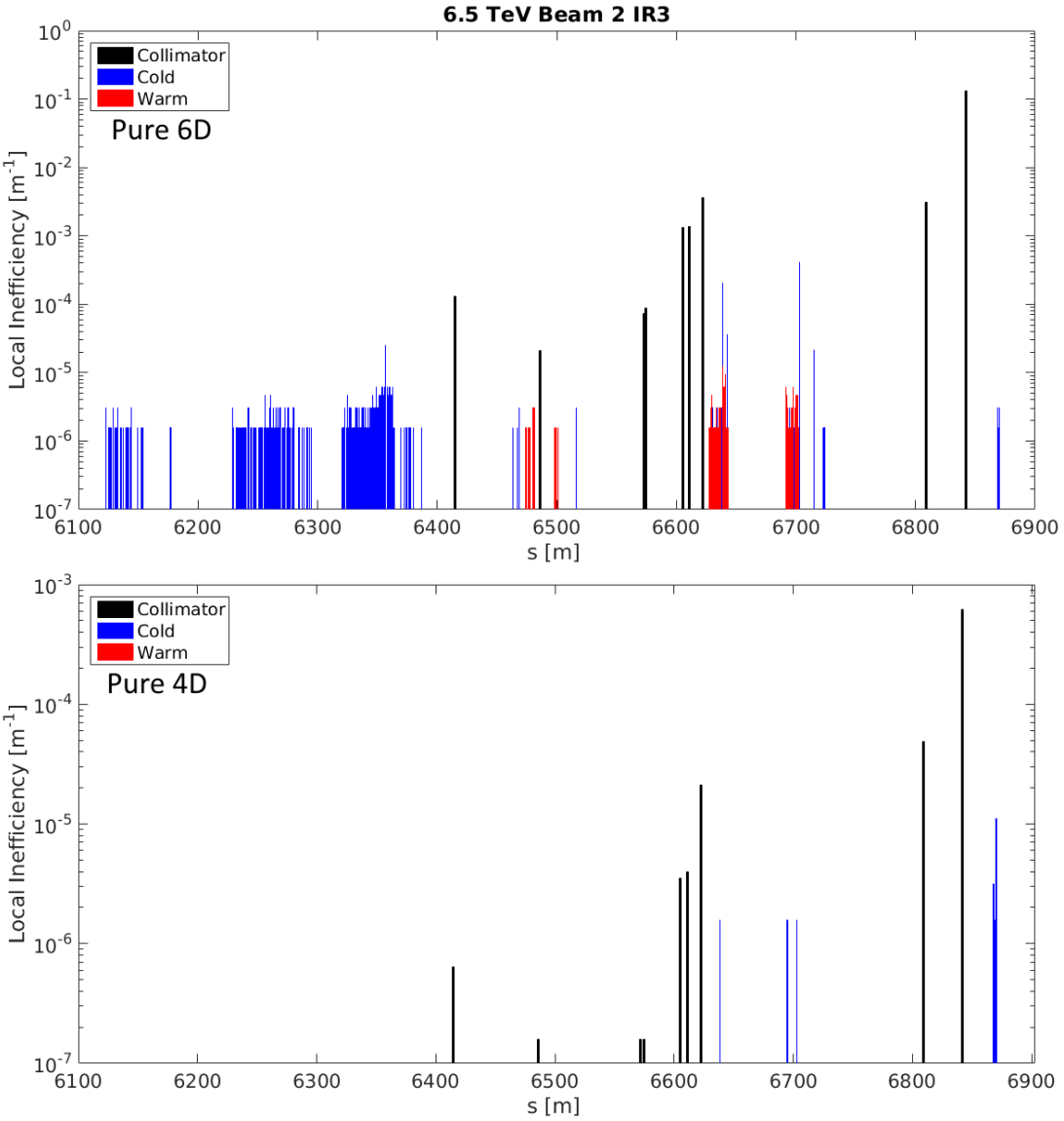

Figure 5.32: Loss maps for beam 2 of the 6.5 TeV LHC in the momentum collimation insertion, plotted in the s co-ordinates for beam 1, using pure collimator materials and comparing the use of 6D tracking (above) with the use of 4D tracking (below). We observe that when using 6D tracking the total number of losses in IR3 are increased by a factor of 200.

conserving `SYMPLECTIC` tracker. The number of losses and their locations is compared in Table 5.6, which shows that both integrator sets provide similar loss distributions with the only difference a factor of 1.5 increase in IR3 collimator losses when using

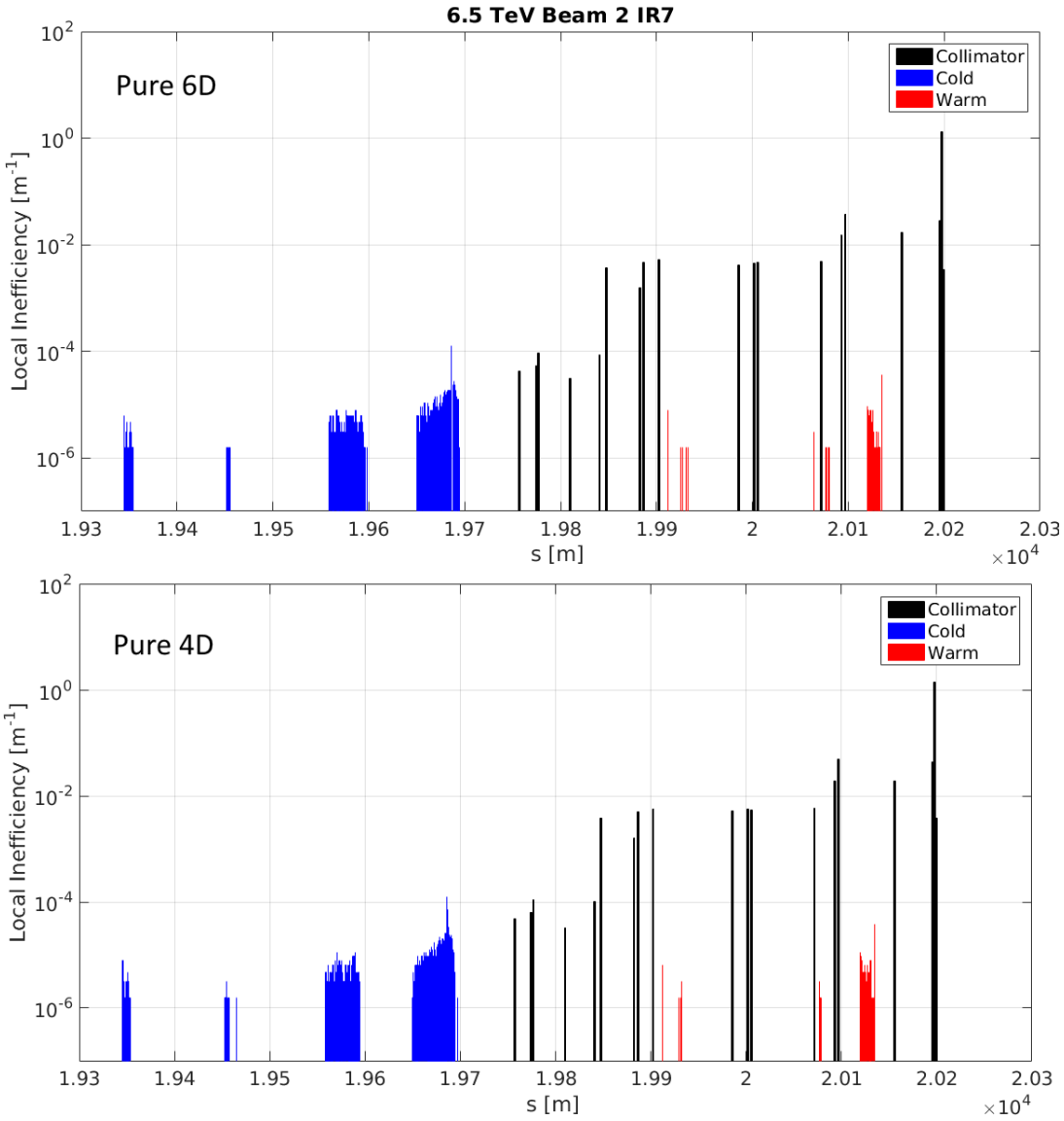

Figure 5.33: Loss maps for beam 2 of the 6.5 TeV LHC in the betatron collimation insertion, plotted in the s co-ordinates for beam 1, using pure collimator materials and comparing the use of 6D tracking (above) with the use of 4D tracking (below). Though not evident, collimator losses are decreased by 7.5% when using 6D tracking.

the TRANSPORT tracker due to the thin lens RF cavity integrator. As expected the betatron collimation insertion losses are nearly identical between the two trackers.

Figure 5.35 shows the transverse distribution of particles that impact upon the most loaded secondary collimator in IR7 when using 4D `SYMPLECTIC` tracking, 6D

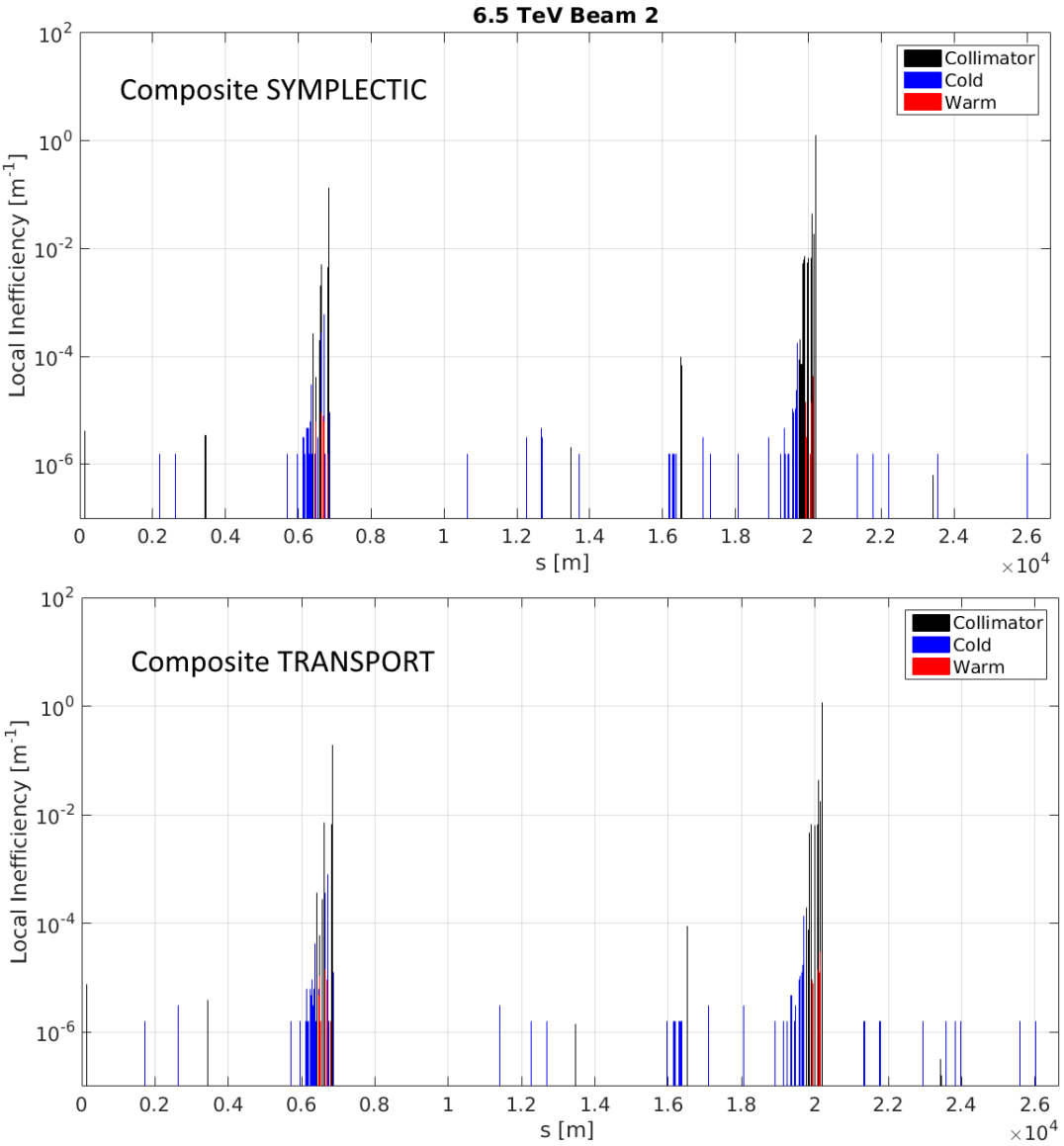

Figure 5.34: Loss maps for beam 2 of the 6.5 TeV LHC, plotted in the s co-ordinates for beam 1, using composite collimator materials and comparing the use of `TRANSPORT` tracking (below) with the use of `SYMPLECTIC` tracking (above). There is no significant difference when using different trackers.

`SYMPLECTIC` is shown in Fig. 5.36, and 4D `TRANSPORT` tracking is shown in Fig. 5.37. The distributions are slightly different when comparing 4D and 6D tracking - in the 4D case the left jaw has more impacts, and the opposite occurs in the 6D case. Negative x indicates the outer side of the accelerator, this effect may be due to synchrotron motion

(*i.e.* 6D tracking) causing particle momentum offset, this in turn would cause particles to travel through bending magnets with an increased bending radius, thus being lost on the inside (right) jaw. As these impacts depend on scattering in primary collimators, this particular difference is not significant enough to be of concern.

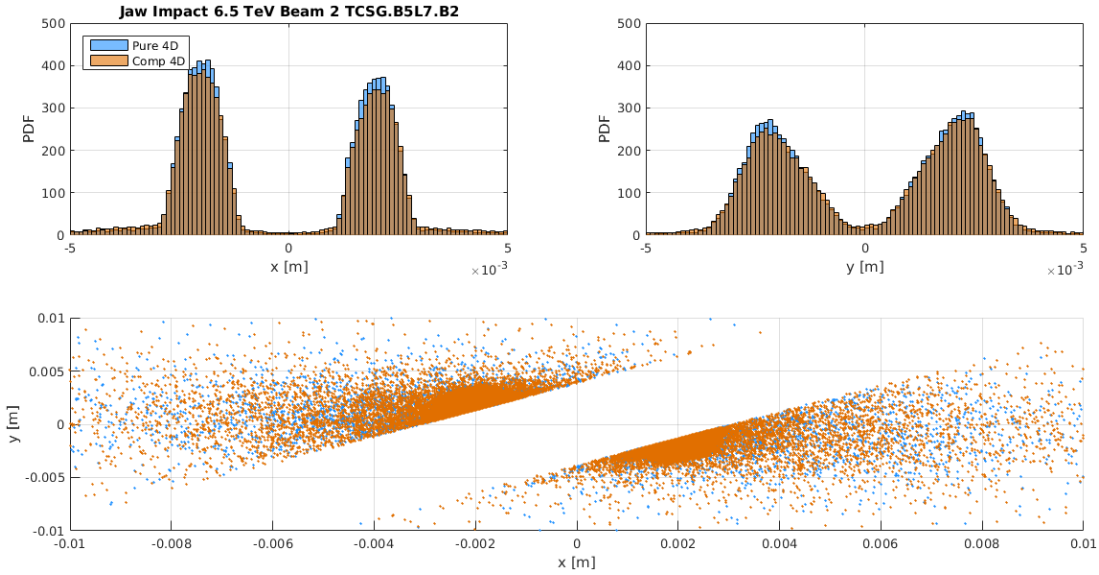

Figure 5.35: Transverse distribution of particles impacting the secondary collimator TCSG.B5L7.B2 for beam 2 of the 6.5 TeV LHC, comparing pure with composite materials when using 4D `SYMPLECTIC` tracking. This collimator has a tilt angle of 2.47 radians. More protons are lost on the outside jaw when using 4D tracking.

5.5.2 Effect of Composite Materials

We may use pure elements to model the LHC collimator jaws in order to perform proton scattering in said jaws. To improve upon this first approximation and benefit from the advanced scattering in MERLIN, the `CompositeMaterial` class and its appropriate handling in the collimation process has been added. We replace the pure elements of carbon, copper, and tungsten, with the composites corresponding to the physical materials used for LHC collimators - CFC AC150K, Glidcop, and Inermet180 respectively. We have already identified that these changes have an effect on proton scattering in a 0.5 m collimator in the previous sections, here we observe how this changes loss maps on a much larger scale.

Figure 5.38 compares the full 6.5 TeV beam 2 loss map using pure collimator materials, with that using composite materials. We observe similar primary and larger

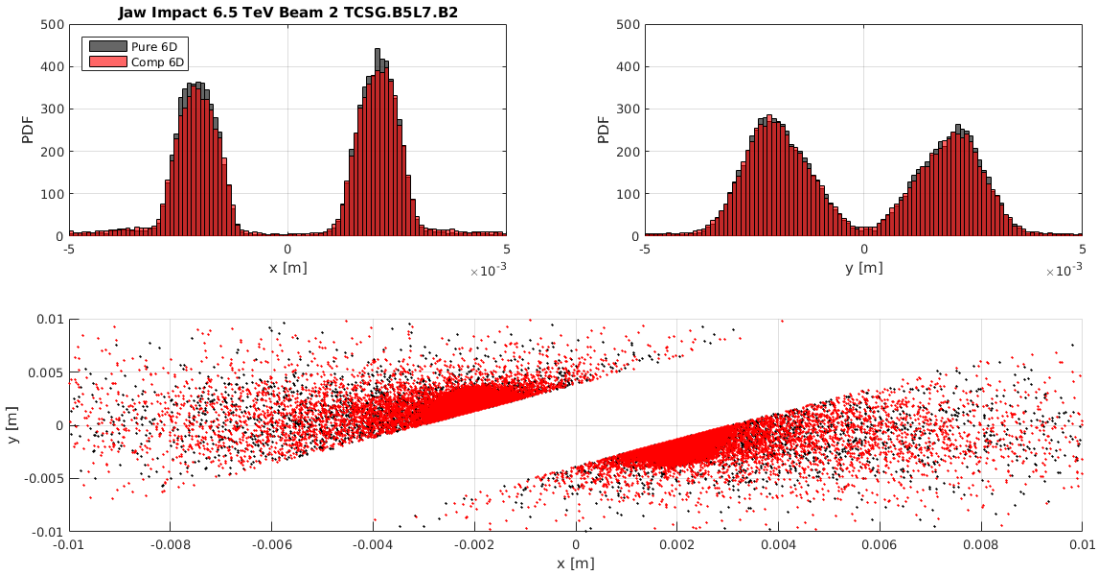

Figure 5.36: Transverse distribution of particles impacting the secondary collimator TCSG.B5L7.B2 for beam 2 of the 6.5 TeV LHC, comparing pure with composite materials when using 6D `SYMPLECTIC` tracking. This collimator has a tilt angle of 2.47 radians. More protons are lost on the inside jaw when using 6D tracking, this is as expected as particles with a momentum offset due to scattering will take a smaller bending radius through dipole magnets and be lost on the inside jaw.

secondary collimator losses when using composite materials, and a 30% increase in cold losses. IR7 is shown in Fig. 5.39, when using composites the losses in secondary collimators and absorbers are slightly increased, as are cold DS losses. Figure 5.40 shows the momentum collimation insertion, from which it appears there is little difference between the two material methods, however there is an increase in collimator losses when using composite materials, as shown in Table 5.7, which compares the losses in the two collimation insertions. This shows that there is a 50% increase in secondary and cold losses, and a 7% increase in primary losses in IR3.

We compare the distribution of losses in the primary horizontal collimator in IR7 with that recorded in SixTrack [95], in Figures 5.41 to 5.44. In Fig. 5.41 we see that the longitudinal distribution of losses is similar in all cases, with a slight reduction in immediate losses as we move from 4D to 6D tracking. The 4D case is most similar to SixTrack in behaviour, with SixTrack having more losses in total, this is likely due to the higher interaction cross section. The loss distribution in x is shown in Fig. 5.42 for both jaws, the difference between the 4D tracking and SixTrack is due to the difference in initial distribution, however it is interesting to note that the 6D losses have a

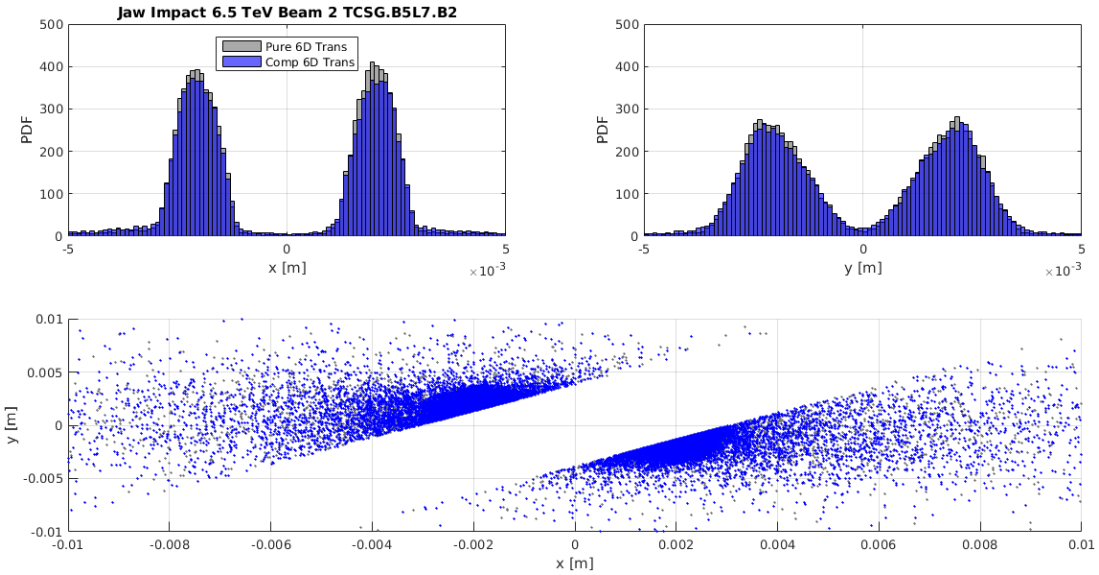

Figure 5.37: Transverse distribution of particles impacting the secondary collimator TCSG.B5L7.B2 for beam 2 of the 6.5 TeV LHC, comparing pure with composite materials when using 4D `TRANSPORT` tracking. This collimator has a tilt angle of 2.47 radians. Again using 6D tracking gives more losses on the inside jaw as expected.

Region	Type	Losses			
		4D Pure	4D Composite	6D Pure	6D Composite
LHC	**Collimator**	6298362	6288308	6391014	6388154
	Cold	3010	3897	3553	4860
	Warm	240	425	425	663
IR7	**Collimator**	6294407	6282753	5819657	5771352
	(Primary)	5493499	5314820	5156956	4934111
	(Secondary)	798662	963148	660756	832726
	Cold	2933	3816	2463	3230
	Warm	240	424	202	350
IR3	**Collimator**	2824	3320	570358	614758
	Cold	4	6	1030	1537
	Warm	0	1	223	313

Table 5.7: Breakdown of losses for beam 2 of the 6.5 TeV LHC comparing simulations using pure and composite materials.

larger variance, which is likely due to an increased impact parameter as off-momentum protons not captured by the momentum collimation insertion are collimated by the horizontal primary. The x' loss distribution in this collimator (Fig. 5.43) is similar in all cases, though it is clear that the 4D MERLIN simulation is closest to SixTrack,

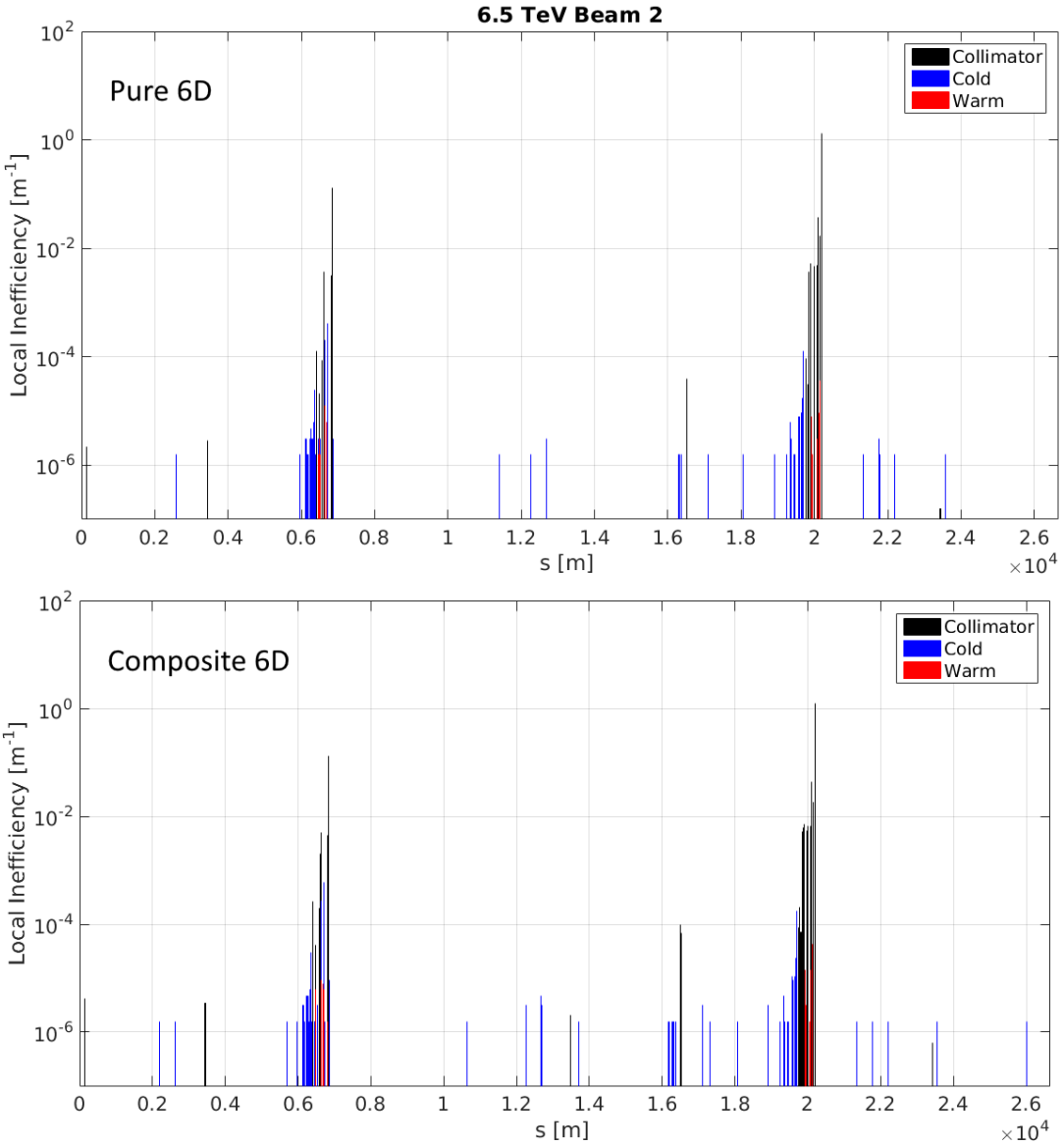

Figure 5.38: Loss maps for beam 2 of the 6.5 TeV LHC, plotted in the s co-ordinates for beam 1, comparing the use of pure materials (above) with composite materials (below) for collimator jaws. Using composites increases cold losses across the LHC, particularly in dispersion suppressor regions post collimation.

the 6D simulation produces a slightly larger variance again due to off-momentum particles being scattered to larger angles. Figures 5.45 and 5.44 show that the y and y' distributions of lost particles are similar for all cases.

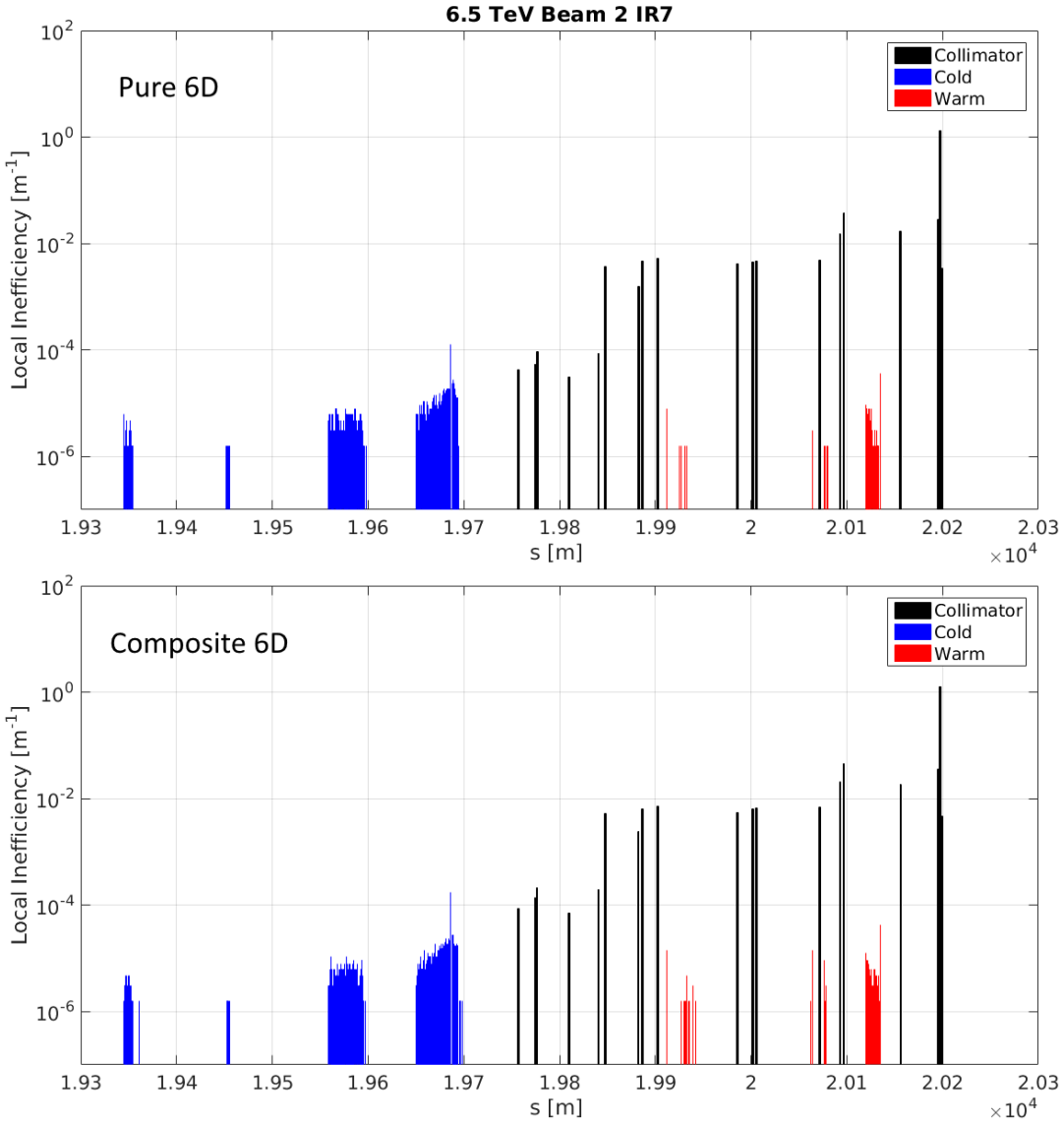

Figure 5.39: Loss maps for beam 2 of the 6.5 TeV LHC in the betatron collimation insertion, plotted in the s co-ordinates for beam 1, comparing the use of pure materials (above) with composite materials (below) for collimator jaws. There is in increase in secondary collimator and cold losses when using composites.

Figure 5.46 shows the longitudinal loss distribution for the most loaded secondary collimator in this loss map. Comparing the 4D and 6D tracking we observe that this has less of an effect than the method of collimator jaw material treatment. More particles are lost in this secondary collimator when using composite materials. This is due to

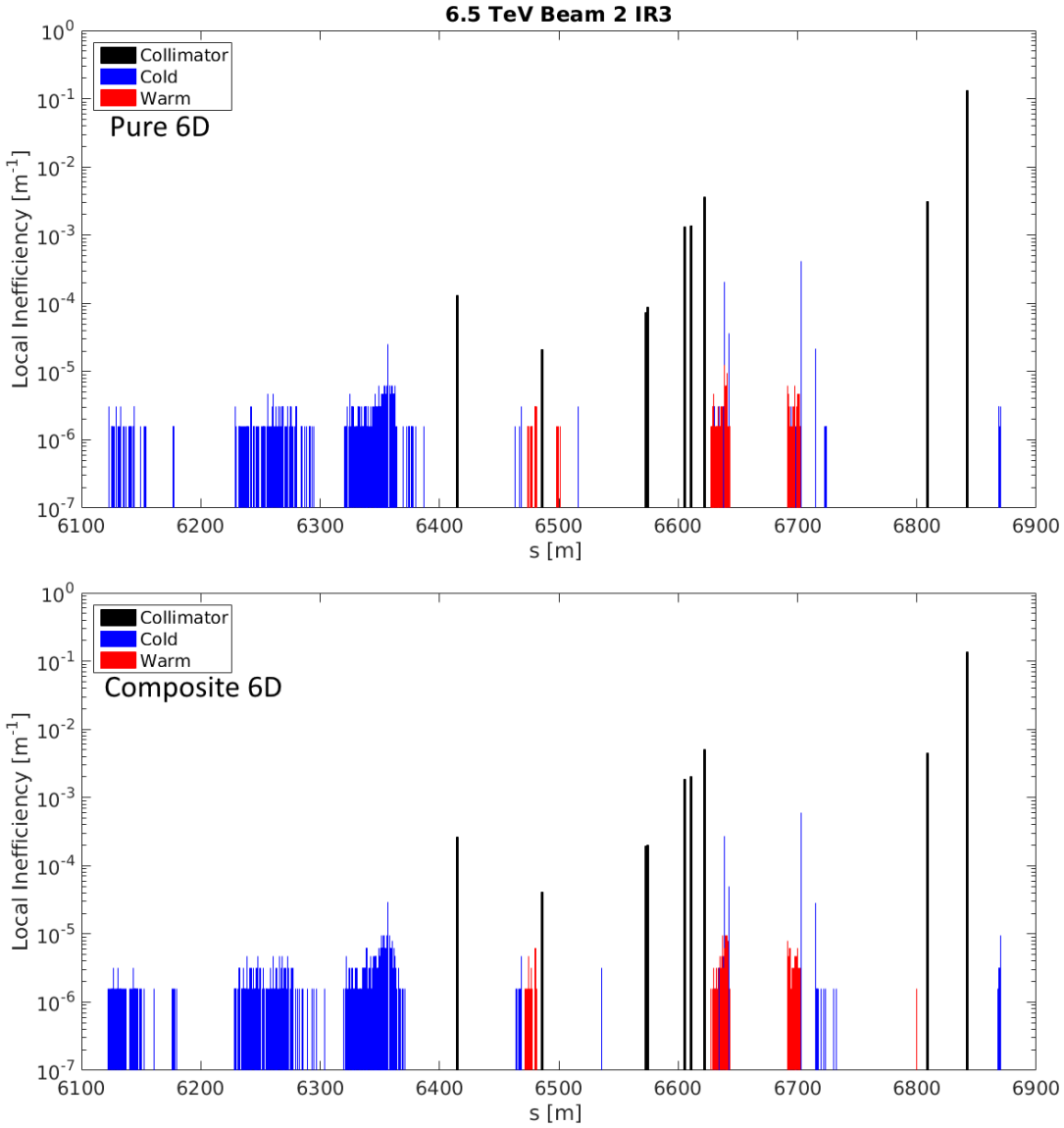

Figure 5.40: Loss maps for beam 2 of the 6.5 TeV LHC in the momentum collimation insertion, plotted in the s co-ordinates for beam 1, comparing the use of pure materials (above) with composite materials (below) for collimator jaws. There is in increase in collimator and cold losses when using composites.

more particles escaping the primary collimators and populating the secondary halo which interacts with the secondary collimators.

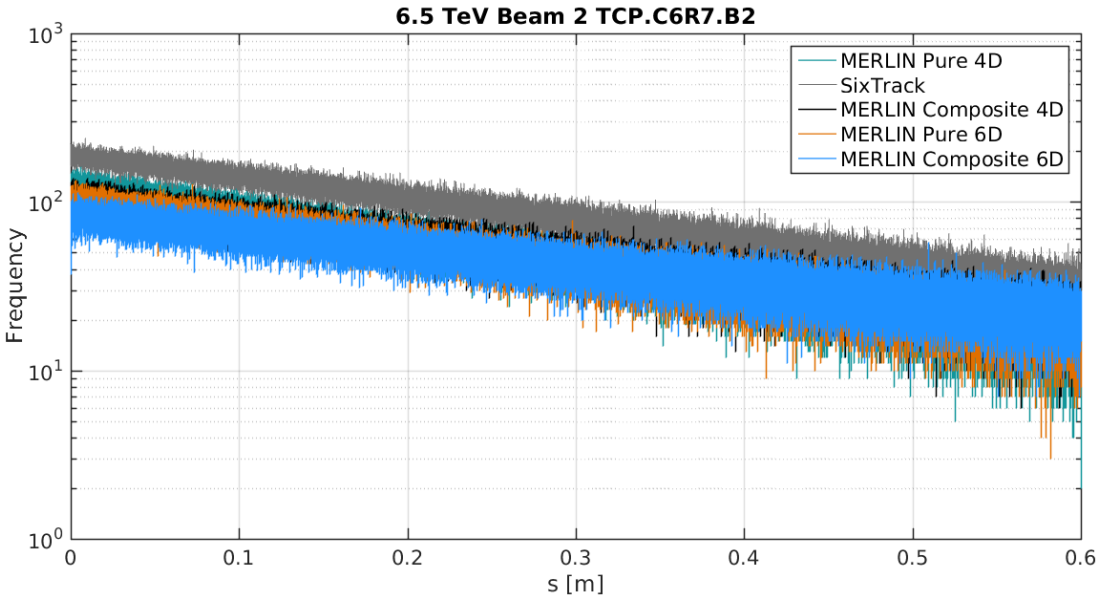

Figure 5.41: Longitudinal distribution of lost particles in the primary collimator TCP.C6R7.B2 for beam 2 of the 6.5 TeV LHC, comparing 4D with 6D tracking, pure with composite materials, and MERLIN results to SixTrack [95]. Use of composites gives a more even spread of inelastic interactions along the length of the collimator. 4D tracking gives more losses nearer the collimator jaw face, and the number of losses in MERLIN is always less than in SixTrack.

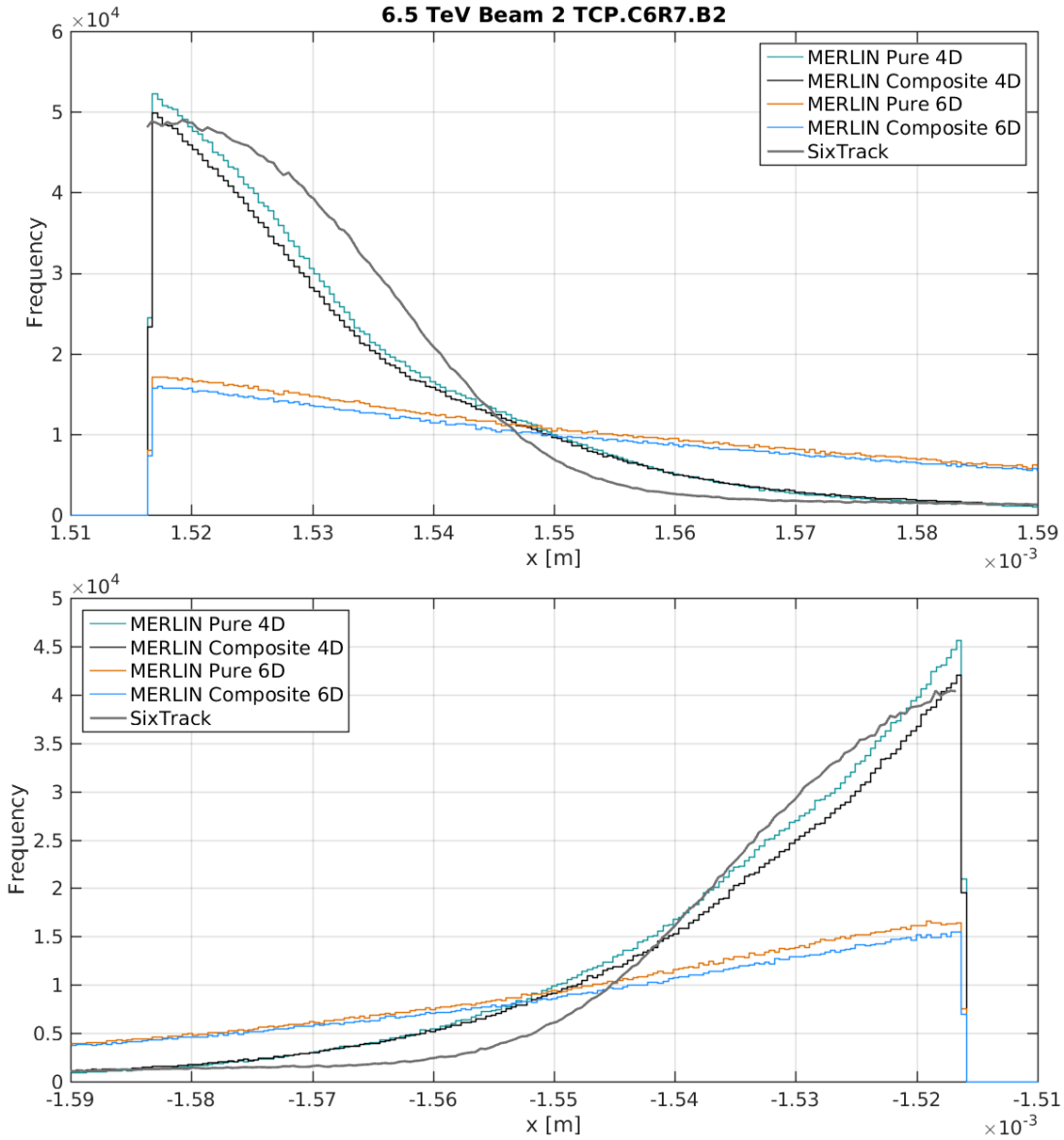

Figure 5.42: x distribution of lost particles in the primary collimator TCP.C6R7.B2 for beam 2 of the 6.5 TeV LHC, comparing 4D with 6D tracking, pure with composite materials, and MERLIN results to SixTrack [95]. The above plot shows the positive jaw, the below plot shows the negative jaw. The higher number of losses closer to the jaw edge in both Sixtrack and 4D MERLIN tracking may be due to a difference in impact parameter on the jaw due to synchrotron motion in 6D tracking. It is difficult to compare MERLIN and SixTrack due to the difference in initial distribution as shown in Fig. 5.27.

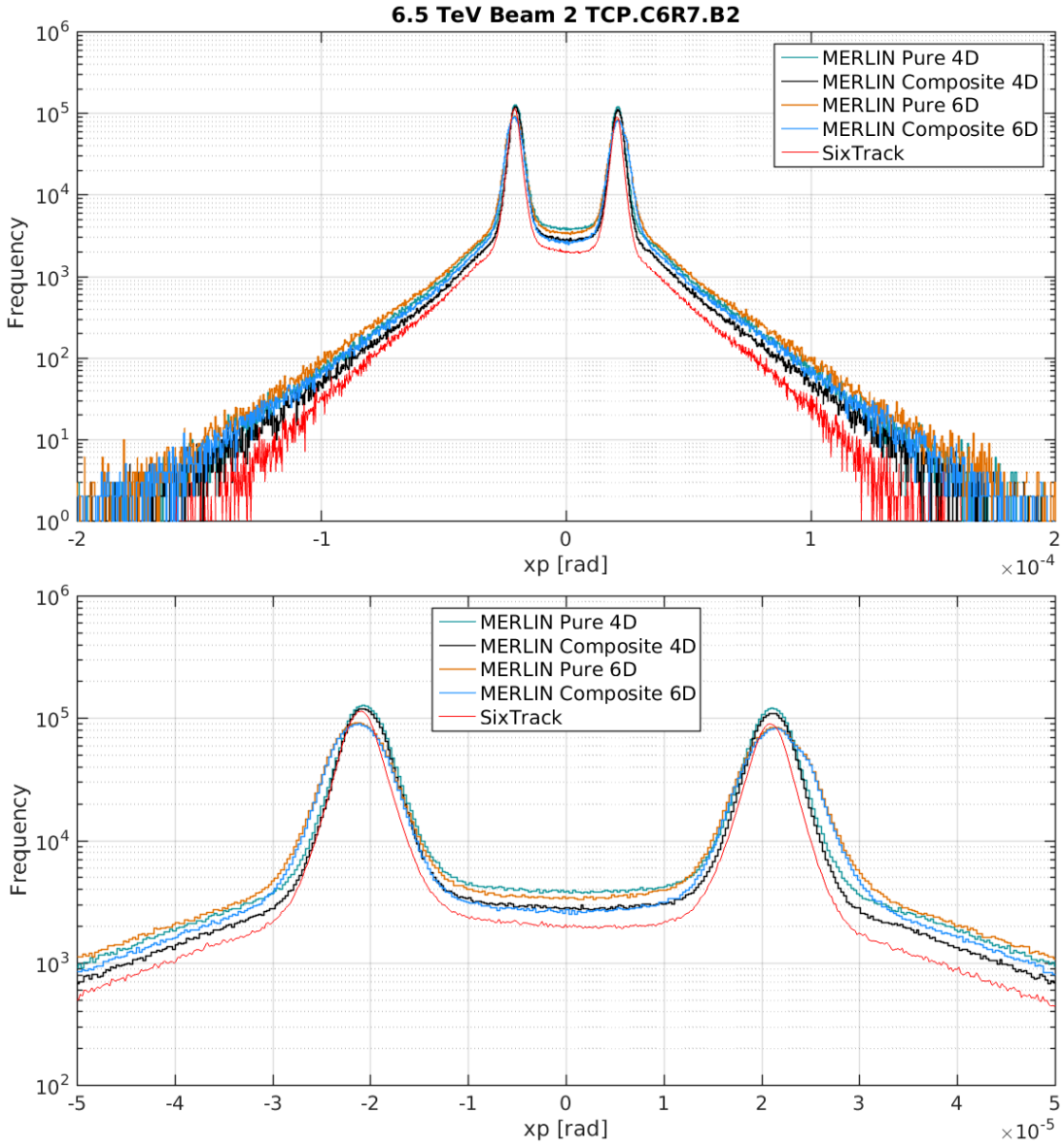

Figure 5.43: x' distribution of lost particles in the primary collimator TCP.C6R7.B2 for beam 2 of the 6.5 TeV LHC, comparing 4D with 6D tracking, pure with composite materials, and MERLIN results to SixTrack [95]. The bottom plot is an enlargement of the peaks in the top plot. We observe that using 6D tracking changes the x' distribution.

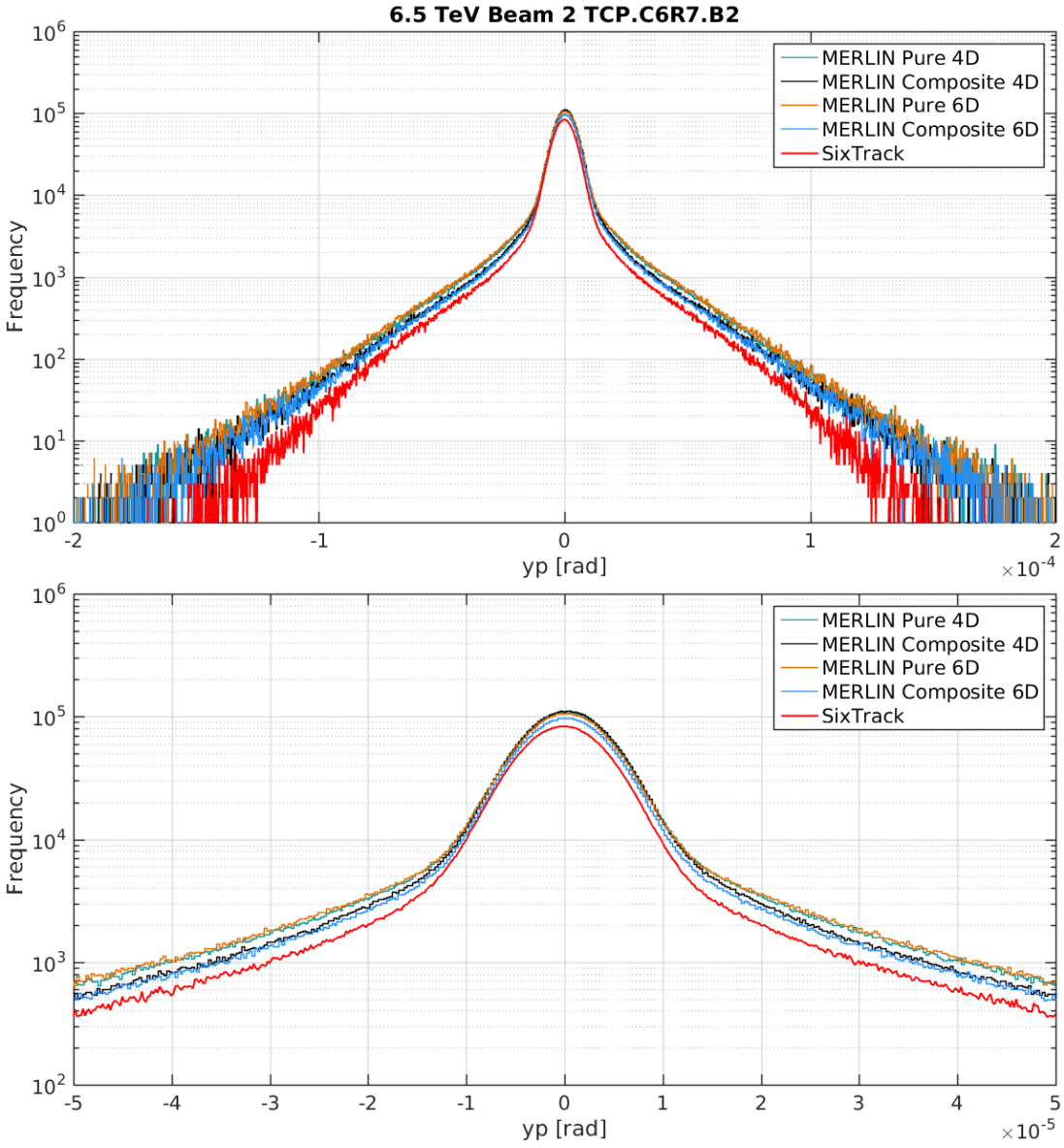

Figure 5.44: y' distribution of lost particles in the primary collimator TCP.C6R7.B2 for beam 2 of the 6.5 TeV LHC, comparing 4D with 6D tracking, pure with composite materials, and MERLIN results to SixTrack [95]. The bottom plot is an enlargement of the peak in the top plot.

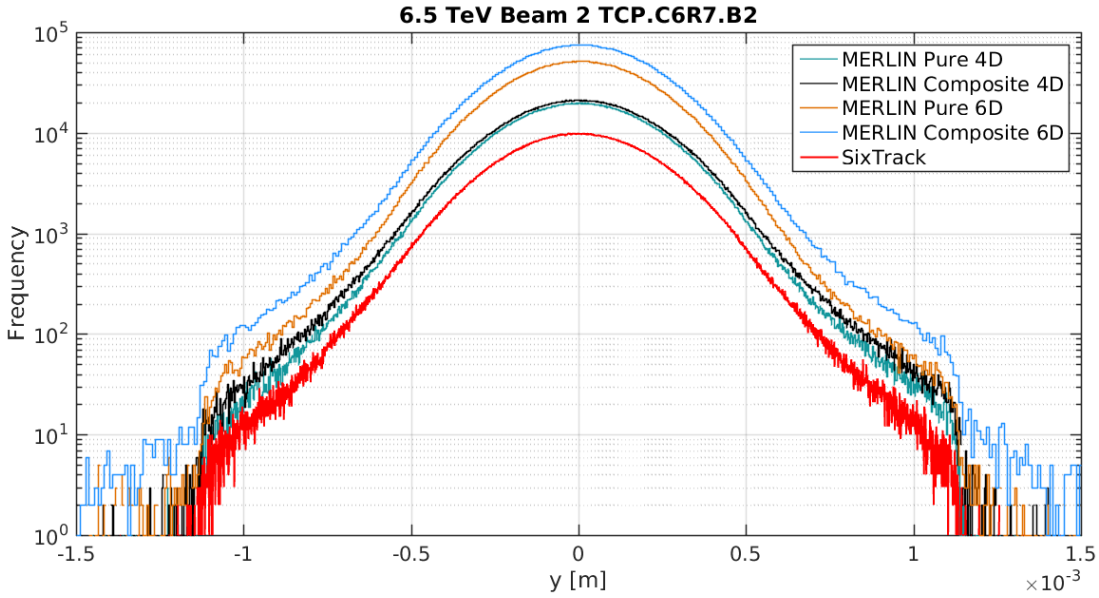

Figure 5.45: y distribution of lost particles in the primary collimator TCP.C6R7.B2 for beam 2 of the 6.5 TeV LHC, comparing 4D with 6D tracking, pure with composite materials, and MERLIN results to SixTrack [95].

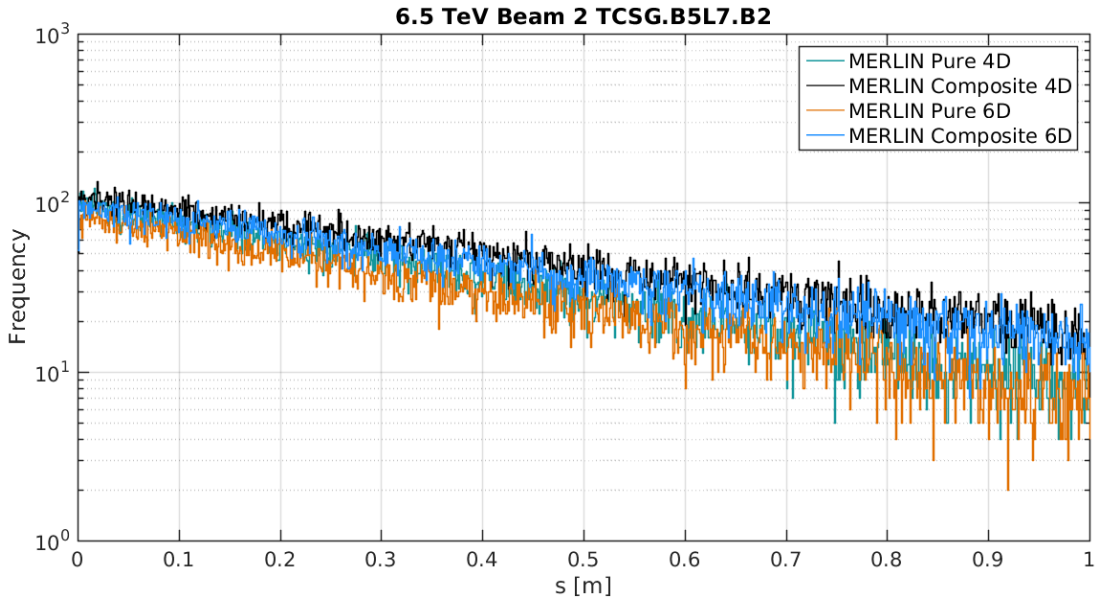

Figure 5.46: Longitudinal distribution of lost particles in the secondary collimator TCSG.B5L7.B2 for beam 2 of the 6.5 TeV LHC, comparing 4D with 6D tracking, and pure with composite materials. This collimator has a tile angle of 2.47 radians.

5.5.3 Comparison with Measurement

A complete quantitative comparison with measured loss maps is not available when using MERLIN or SixTrack, and a simulation of full proton scattering and secondary shower production for the entire LHC would be immensely computationally expensive. We may use the approximations of MERLIN to predict loss location and magnitude. Figure 5.47 compares the measured LHC loss map with 4D and 6D MERLIN simulations using pure elements for collimator materials. The collimation hierarchies in IR7 and IR3 are observed in all loss maps.

Measured loss maps are generated using an excitation of a single beam's betatron amplitude, which forces the loss pattern generated by the betatron collimation insertion only. This is shown by the magnitude of losses in IR3 being of order 10^{-3}, which is similar to that in the 4D simulation. In normal machine operation the momentum collimation insertion is in use as particles lose momentum in elastic collisions at the experimental IPs, as well as mismatches with the RF bucket and other mechanisms discussed in Chapter 3. The loss map is measured with BLMs, and using a single beam allows us to disentangle BLM signals which would be generated from losses in both beams during normal operation. As the absolute loss rate is higher in a qualification loss map, the signal to noise is much higher than in normal operation [71].

IR7 is the limiting loss location for the LHC, and is thus the main concern, as depicted in Fig. 5.48. We note that BLMs are not distributed continuously throughout the LHC, and thus signals appear more concentrated in the collimation insertions where there are more BLMs. The collimation hierarchy is clear in all loss maps, and the location of the highest cold losses - the DS region following the betatron collimation insertion - is similar though secondary particles cause a higher signal in the measured loss map. IR3 is shown in Fig. 5.49, this is mostly dominated by BLM signal from secondary losses, some of which are likely to be from beam 1 (due to the hierarchy direction). Though the highest loss is of similar magnitude to that in the 4D MERLIN simulation, losses are not of concern in the regime of the measured noise.

It is difficult to draw conclusions with direct comparisons between MERLIN (or SixTrack) generated and measured losses. This is due mainly to the effect of secondary showers. The missing step can be performed using a shower code such as FLUKA [19], and such comparisons have been performed. In previous comparisons it has been shown that losses predicted in SixTrack, and propagated in FLUKA to produce secondary particle showers, show good agreement with measured losses [71]. Thus in the next section we compare MERLIN to SixTrack to draw firmer conclusions on the validity of

MERLIN as a loss map generator for the LHC and HL-LHC.

More detailed comparisons of MERLIN and measured losses are currently being performed [98], though these also omit the use of shower simulations. A possible method of direct comparison would be to perform a machine development (MD) experiment at the LHC, using a bunch of low population in only one beam direction to trigger a betatron loss at a specific primary collimator. USing a small number of protons could provide a cleaner loss signal with the BLMs, and knowledge of the loss (*i.e.* which primary collimator is hit - horizontal, vertical, or skew - as well as full collimator settings) would provide the simulation set-up for comparison in MERLIN.

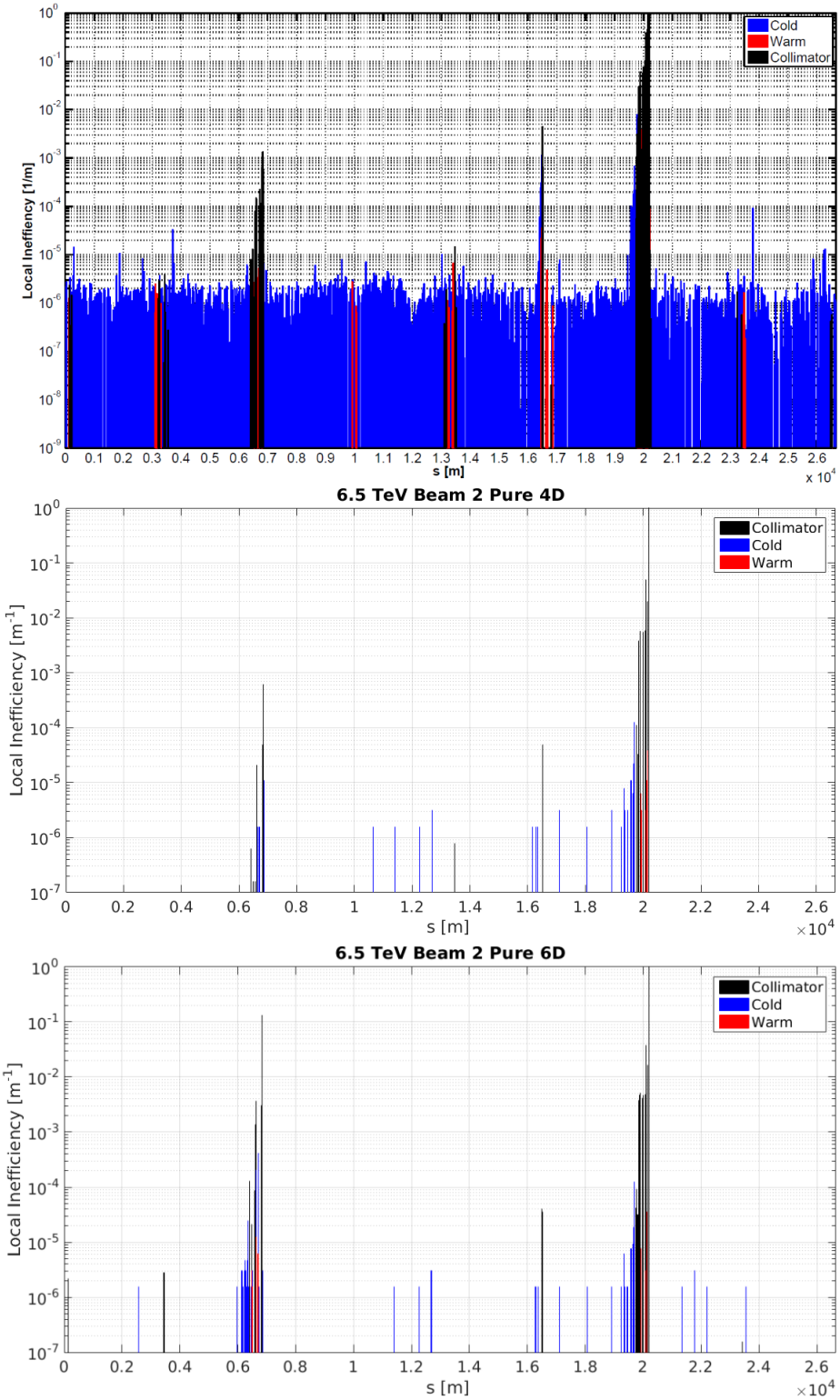

Figure 5.47: Loss maps for beam 2 of the 6.5 TeV LHC, plotted in the s co-ordinates for beam 1, comparing measurement (above) with MERLIN using 4D (middle) and 6D (below) `SYMPLECTIC` tracking, and pure elements for collimator jaws. The main loss locations and relative magnitudes are similar when comparing MERLIN with measurements, with the exception of the IR6 collimator which gives a large loss in measurements. This dump protection collimator is likely closed more than in the MERLIN simulations. This may explain why the MERLIN 6D IR3 loss is much larger than that in measurement, as protons are lost in IR6 rather than downstream in IR3.

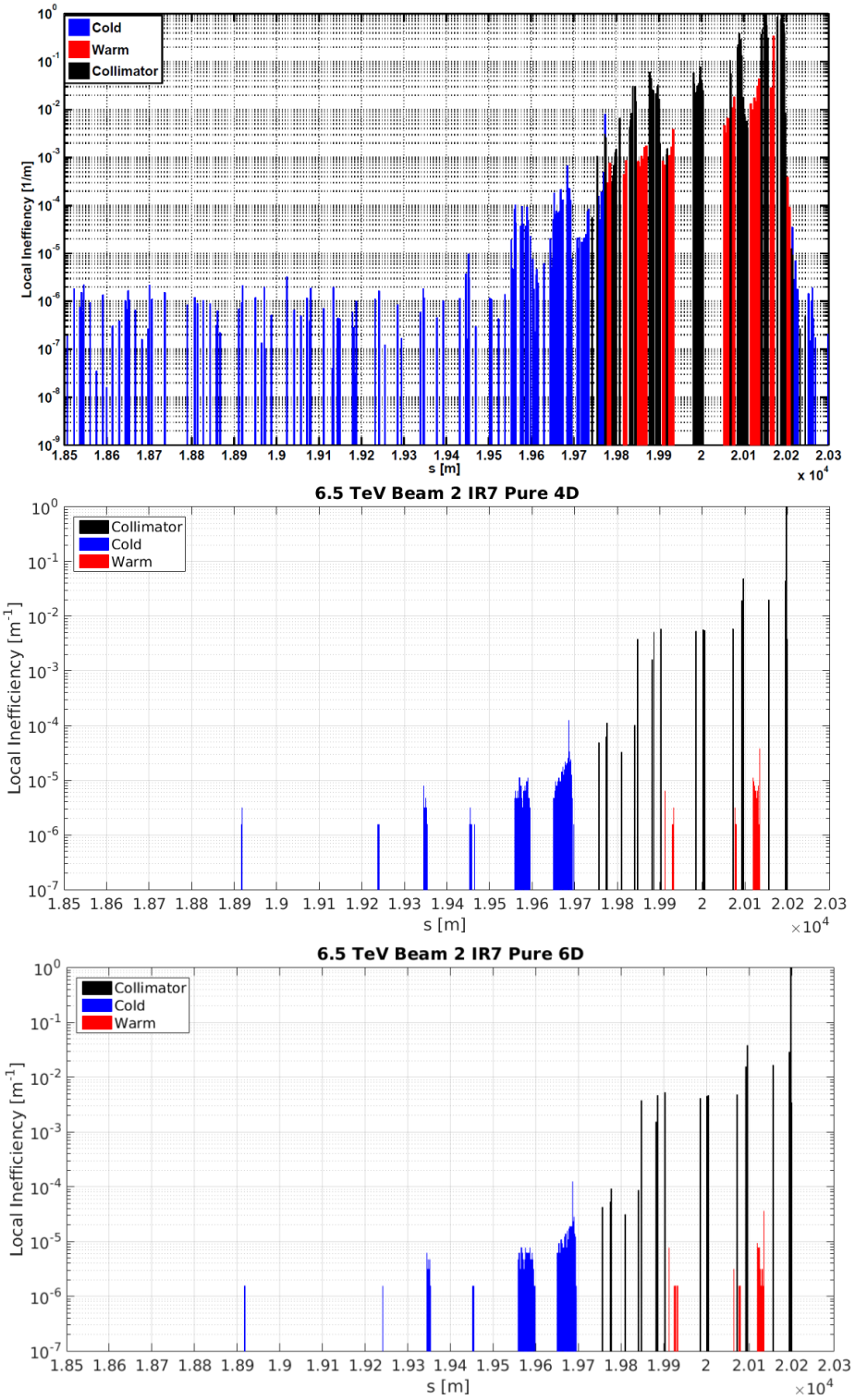

Figure 5.48: Loss maps for beam 2 of the 6.5 TeV LHC in the betatron collimation insertion, plotted in the s co-ordinates for beam 1, comparing measurement (above) with MERLIN using 4D (middle) and 6D (below) `SYMPLECTIC` tracking, and pure elements for collimator jaws.

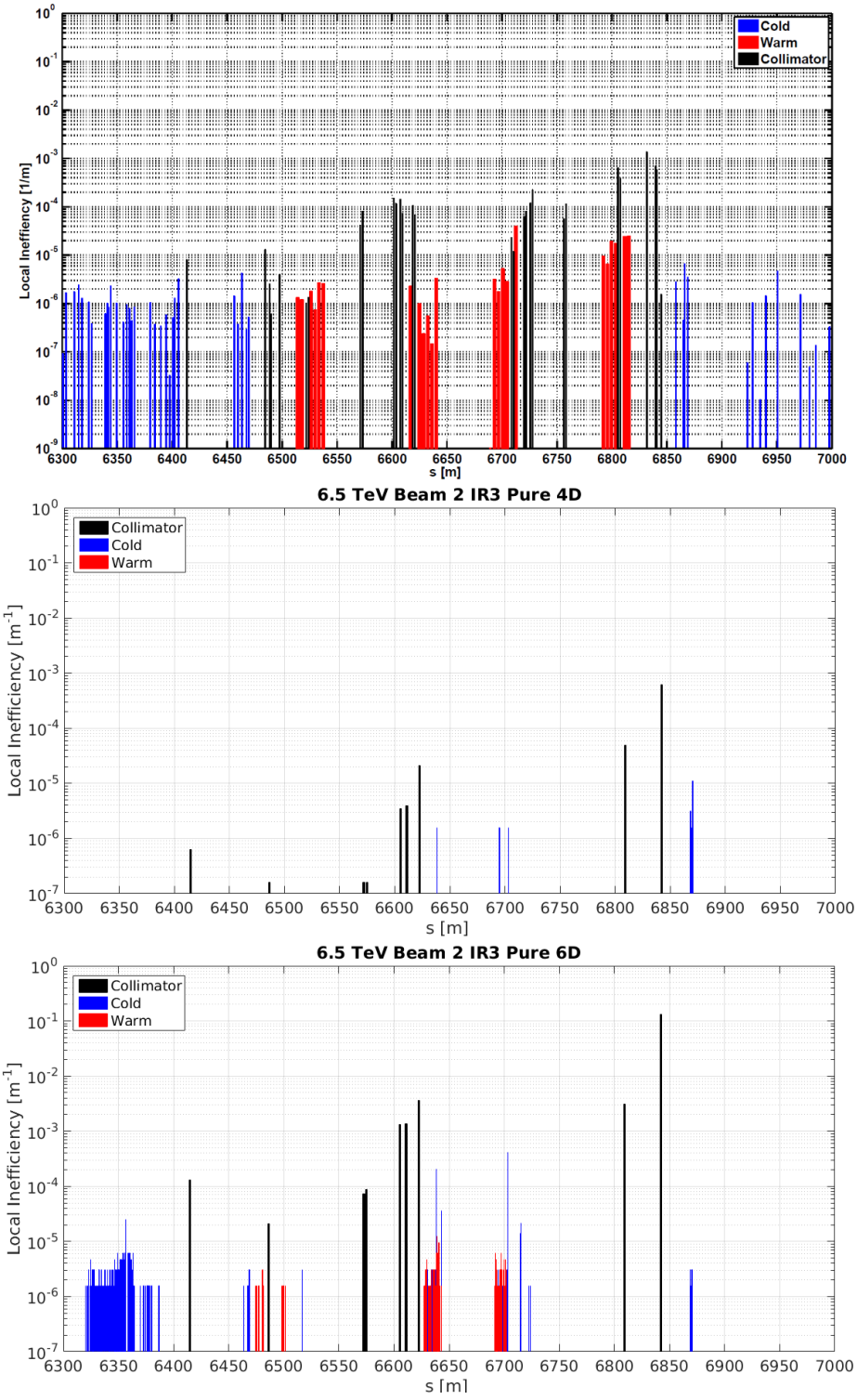

Figure 5.49: Loss maps for beam 2 of the 6.5 TeV LHC in the momentum collimation insertion, plotted in the s co-ordinates for beam 1, comparing measurement (above) with MERLIN using 4D (middle) and 6D (below) `SYMPLECTIC` tracking, and pure elements for collimator jaws.

5.5.4 Comparison with SixTrack

The 6.5 TeV beam 2 simulation at flat top was performed using MERLIN and SixTrack [95] with identical collimator settings in order to directly compare loss maps. SixTrack was run using homogeneous composites, MERLIN with either pure elements or advanced composites, the initial distributions for these simulations are shown in Fig 5.27 to 5.30. Figure 5.50 compares the SixTrack loss map with 4D and 6D MERLIN simulations using pure elements for collimator materials. Upon inspection it appears that SixTrack is using 4D tracking, as the losses in IR3 are of much lower magnitude than the 6D MERLIN case. This may also be due to the difference in initial proton distributions (in transverse real space as shown in Fig. 5.27).

Figure 5.51 compares SixTrack data to the 4D MERLIN simulation using composite collimator materials. Loss locations and magnitudes are almost identical over the entire machine, with the exception of a few small order cold losses which are dependent on the last interaction of a proton with the collimator from which it escaped.

Comparing IR7, Fig. 5.52 shows better agreement between SixTrack and MERLIN (using composites) in terms of collimator loss magnitudes. The location of losses is similar in all cases with a small difference in cold DS losses further from the collimation insertion. The use of homogeneous composite materials in SixTrack leads to a difference in the contribution of each scattering process to the total cross section, and may be the reason for the greater amount of cold losses in SixTrack when compared to the MERLIN composite. It is clear when comparing composite and pure materials in MERLIN, that the pure elements provide an underestimate of cold losses in this region. There are fewer proton interactions with pure materials, so the likelihood of a single diffractive interaction, which provides the large momentum transfer required for a cold DS loss, is reduced.

Figure 5.53 provides a closer look at the DS region post IR7. From this it is clear that when using composite materials more protons are lost in the final collimators in MERLIN as compared to SixTrack, and thus fewer cold DS losses occur. The combination of advanced scattering and composite treatment suggest that SixTrack slightly overestimates the magnitude of these losses. In this case as we are using materials that are dominated by a single pure element it is likely that this difference is mostly due to scattering routines. The pattern of losses is similar in all cases. The magnitude of the highest cold loss (around s = 19690 m) is much larger in MERLIN. It may be that the *dp* cut-off given by the advanced single diffractive scattering routine translates to a more localised concentration of losses in this particular position relative to the proton

interactions with the primary collimator.

The momentum cleaning insertion in IR3 is compared in Fig. 5.54. The losses shown in this case are dependent on the debris from IR7 and thus not as simple to compare directly. The magnitude and location of collimator losses is similar when comparing the MERLIN composite simulation with SixTrack.

Overall the comparison with SixTrack shows that MERLIN is in good agreement, with small differences which are attributed to differences in scattering routines and possibly initial distributions. Thus we can conclude that, as SixTrack has previously been combined with shower simulations to compare favourably with measurements [71], MERLIN provides a valid tool for loss map generation. Table 5.8 compares the distribution of losses in these simulations as a percentage of the total loss. From this we see that the proportion of collimator losses is always higher in MERLIN, and as a result the proportion of cold losses is diminished. The composite simulation provides slightly more cold losses than that using pure materials as expected, and the use of 6D tracking gives rise to a higher proportion of warm losses, which are mostly in the momentum collimation insertion.

Case	**Loss Location**	**Percentage of Total Losses**
SixTrack	Collimator	99.9017
	Cold	0.0927
	Warm	0.0055
MERLIN 4D Pure	Collimator	99.948
	Cold	0.0478
	Warm	0.0038
MERLIN 4D Composite	Collimator	99.9313
	Cold	0.0619
	Warm	0.0068
MERLIN 6D Composite	Collimator	99.9134
	Cold	0.076
	Warm	0.0104

Table 5.8: Percentage of total losses for beam 2 of the 6.5 TeV LHC comparing simulations using pure and composite materials with SixTrack.

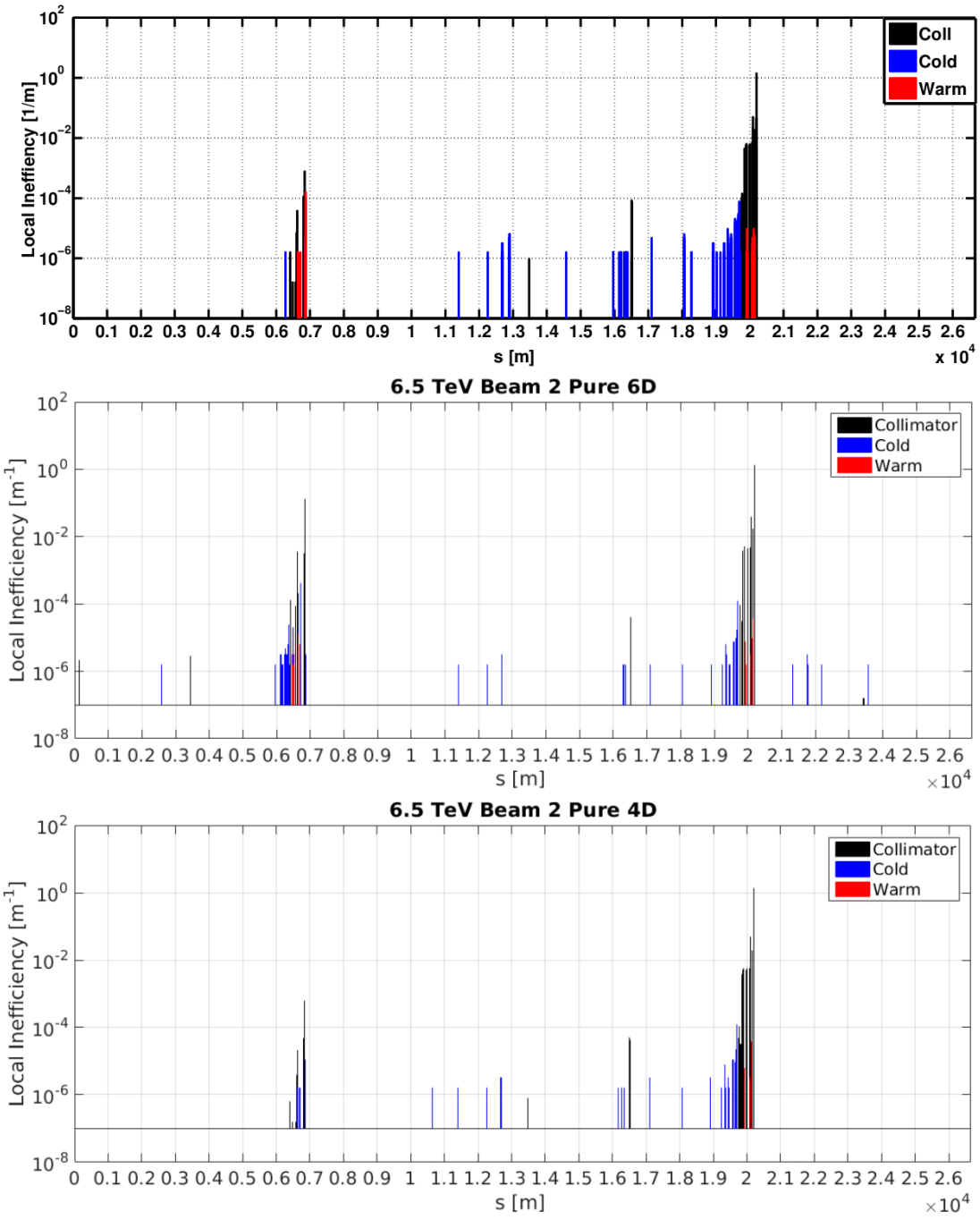

Figure 5.50: Loss maps for beam 2 of the 6.5 TeV LHC in the betatron collimation insertion, plotted in the s co-ordinates for beam 1, comparing SixTrack (above) with with MERLIN using 4D (middle) and 6D (below) `SYMPLECTIC` tracking, and pure elements for collimator jaws. From this comparison it appears that SixTrack is run with 4D tracking as this agrees well with MERLIN 4D tracking.

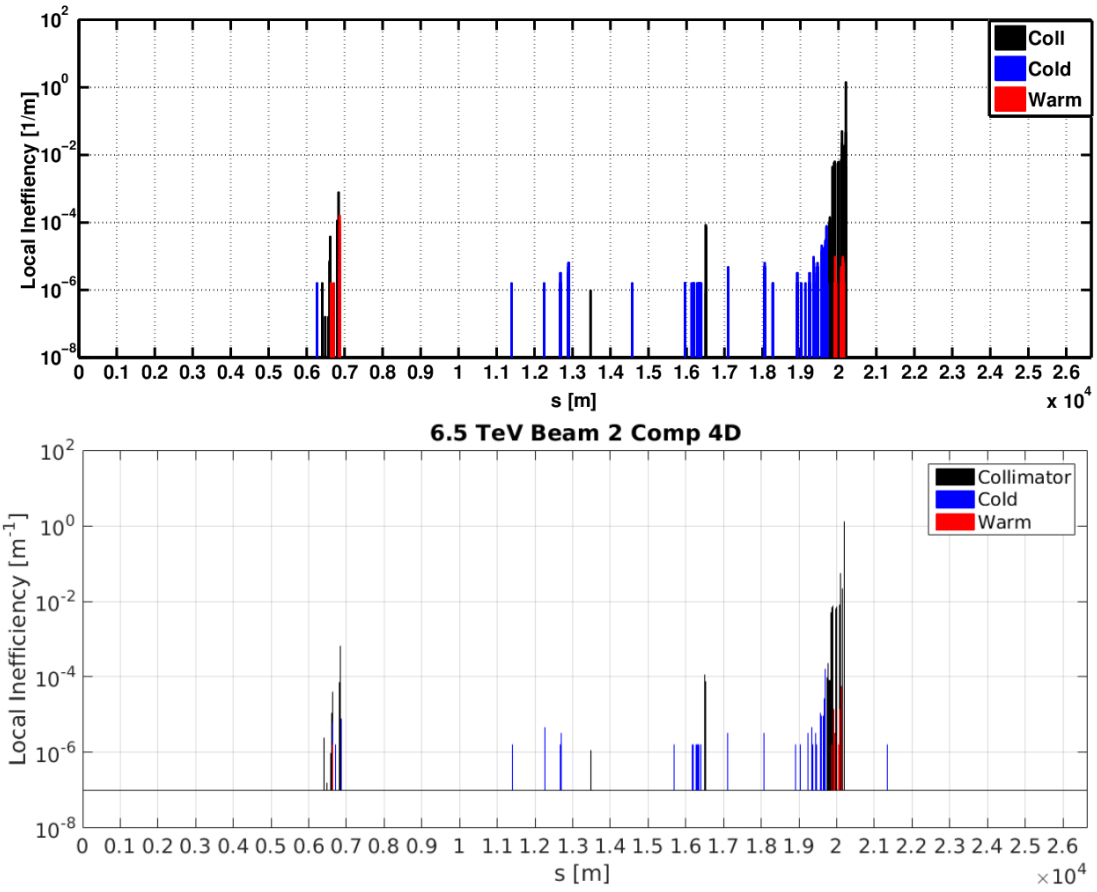

Figure 5.51: Loss maps for beam 2 of the 6.5 TeV LHC in the betatron collimation insertion, plotted in the s co-ordinates for beam 1, comparing SixTrack (above) with with MERLIN using 4D (below) `SYMPLECTIC` tracking, and composite materials for collimator jaws.

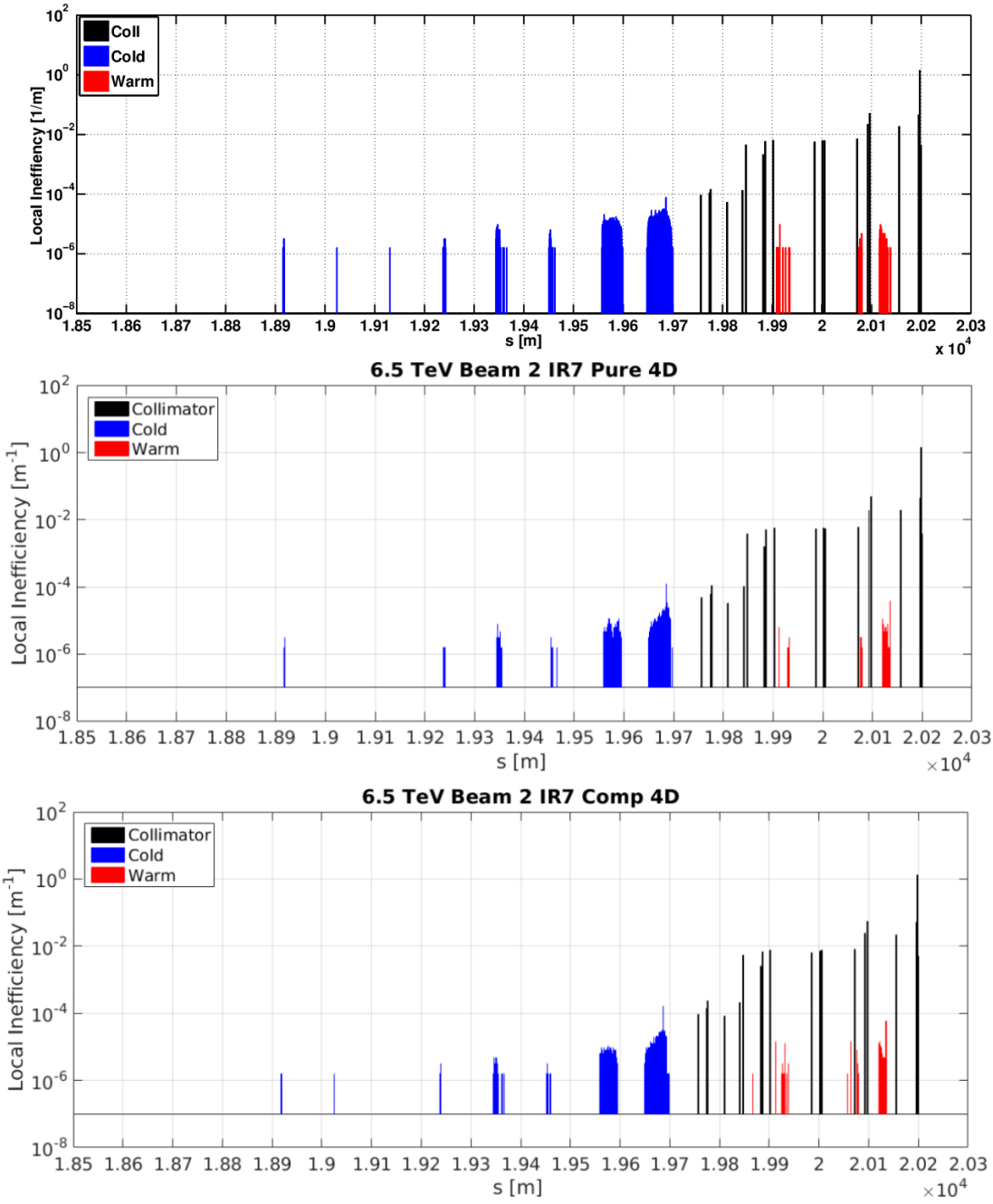

Figure 5.52: Loss maps for beam 2 of the 6.5 TeV LHC in the betatron collimation insertion, plotted in the s co-ordinates for beam 1, comparing SixTrack (above) with with MERLIN using pure elements (middle) and composite materials (below) for collimator jaws, and `SYMPLECTIC` tracking. The larger cold losses in SixTrack are due to the larger single diffractive cross section in SixTrack.

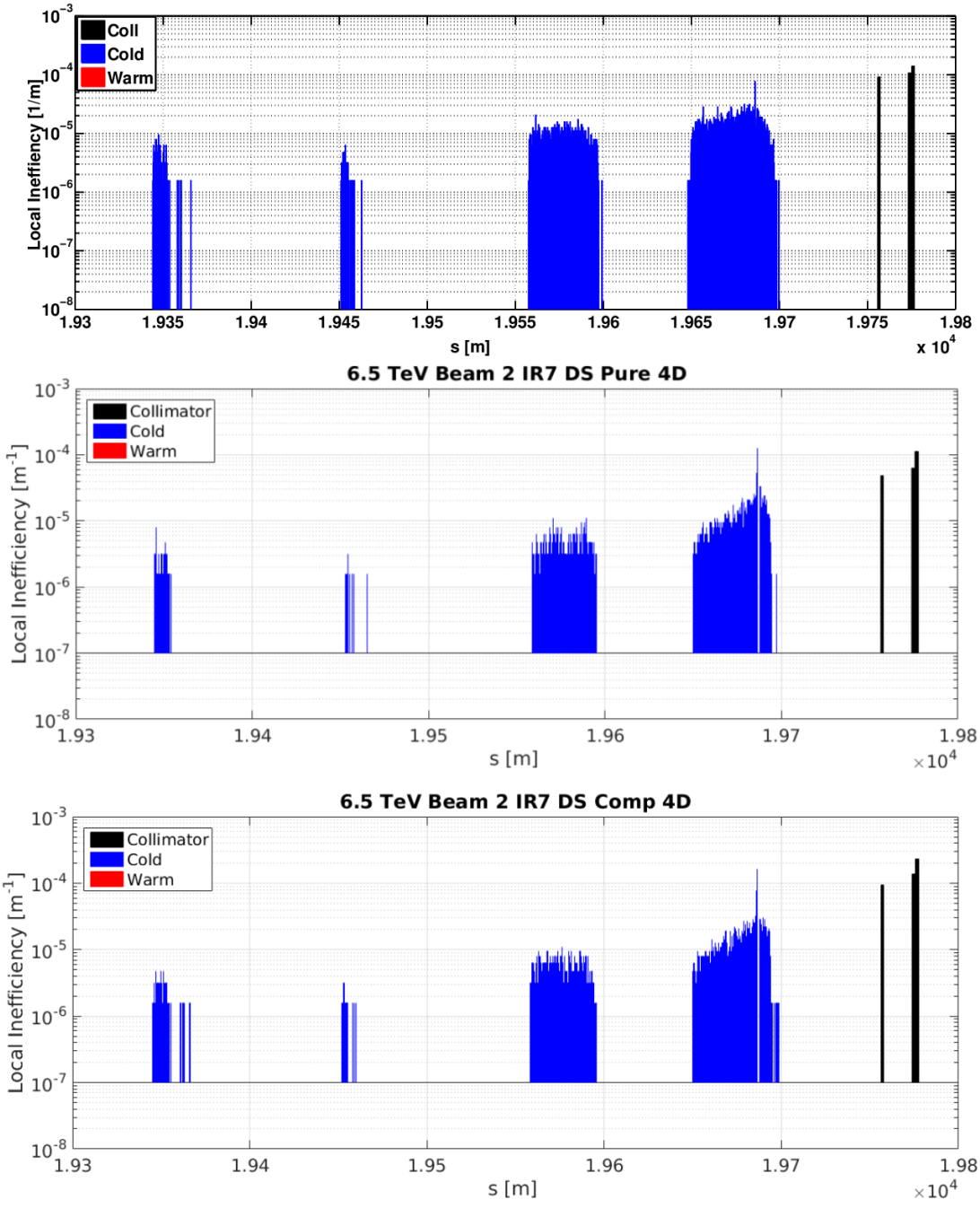

Figure 5.53: Loss maps for beam 2 of the 6.5 TeV LHC in the dispersion suppressor region following the betatron collimation insertion, plotted in the s co-ordinates for beam 1, comparing SixTrack (above) with with MERLIN using pure elements (middle) and composite materials (below) for collimator jaws, and `SYMPLECTIC` tracking. The larger cold losses in SixTrack are due to the larger single diffractive cross section in SixTrack.

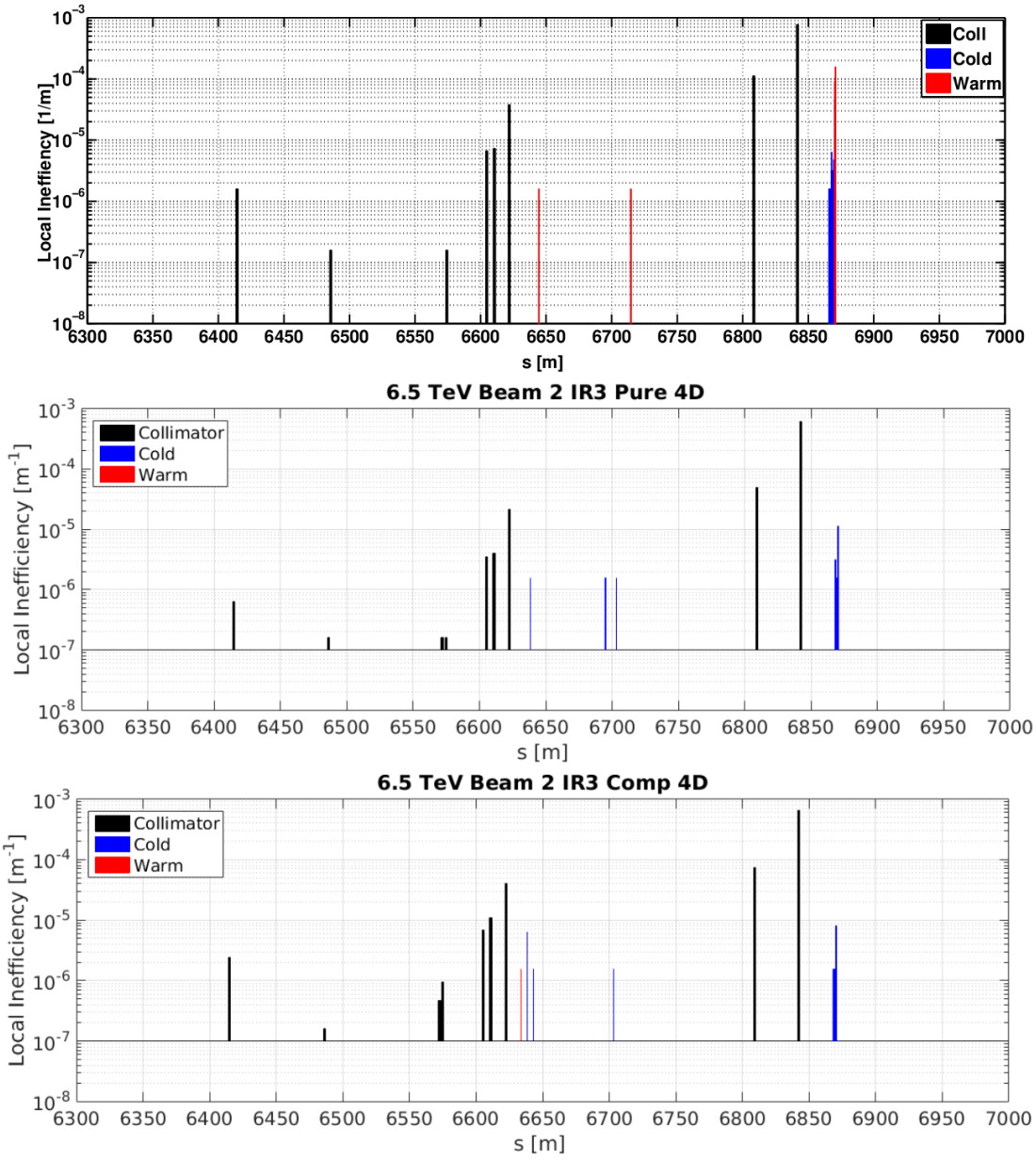

Figure 5.54: Loss maps for beam 2 of the 6.5 TeV LHC in the momentum collimation insertion, plotted in the s co-ordinates for beam 1, comparing SixTrack (above) with with MERLIN using pure elements (middle) and composite materials (below) for collimator jaws, and `SYMPLECTIC` tracking. Loss locations are similar in MERLIN and SixTrack.

5.6 HEL

5.6.1 SixTrack Comparison

In order to benchmark the `MERLIN` HEL process, the test case of the Tevatron HEL hardware in the nominal LHC (lattice v.6.503) was repeated as in [68]. In this case the parameters used were those in the first column of tables 3.9 (with the caveat that the active length $L = 2\ m$) and 3.10, all simulations in this section use these parameters unless otherwise stated.

5.6.1.1 SixTrack HEL Radius Error

The SixTrack implementation of the HEL exists as the 'elens' subroutine of collimation. The HEL is implemented as a special collimator, if a proton hits the HEL, the `collimate_elens` function is called, which in turn calls the `elens_kick` function to perform the HEL kick on the proton.

To use the HEL in SixTrack a special block is required in the collimator input file, in which a number of parameters are defined by the user. These include; the HEL inner radius R_{min} in σ, the ratio $g = \frac{R_{max}}{R_{min}}$, the current I, voltage V, and the operation mode.

In order to fully benchmark MERLIN's HEL implementation, the SixTrack implementation had to be resurrected. SixTrack was previously widely spread, with different research groups using separate offshoots. For example the collimation version includes scattering (based on the K2 code [3]), and post-processed aperture checking. The version of SixTrack that includes the HEL was written for crystal and HEL collimation by Valentina Previtali [68]. The crystal routines were passed on to, and comprehensively updated by Daniele Mirarchi [22]. The HEL routines were not used and were subsequently neglected.

After obtaining Valentina's original version of SixTrack, it was compiled and run for the nominal LHC test case. As a first test the Poincaré section for a DC HEL in the nominal LHC was plotted. The effect of the HEL on the bunch in xx' phase space was not observed. After investigation it was clear that the value for the minimum HEL radius R_{min}, and thus the maximum HEL radius R_{max} were not correct in the HEL routine.

The R_{min} value is fed from the collimator definition input file as mentioned previously. It is entered in units of beam σ, and together with the angle of the collimation plane in transverse xy space (with respect to the horizontal), it is used to set the radii

at a given value of σ, where σ is the corresponding beam size at the given angle. For example if the angle given is 0 rad, $\sigma = \sigma_x$, and $R_{min} = n\ \sigma_x$, where n is the value input by the user.

At the chosen location of the HEL in the nominal LHC (s = 10037 m), $\sigma_x = 293 \cdot 10^{-6}$ m, or 293 μ m. R_{min} is defined as 4 σ_x and R_{max} as 6.8 σ_x as stated in Table. 3.10, thus in SixTrack the user must enter $n = 4$, and $g = 1.7$. These values pass through the main tracking interface of SixTrack, then the collimation routine, before finally reaching the elens subroutine. When output at the elens routine the values returned in metres were $R_{min} = 2.834 \cdot 10^{-3}$ m, $R_{max} = 4.8 \cdot 10^{-3}$ m, which are a factor ≈ 4.2 larger than the expected $R_{min} = 1.17 \cdot 10^{-3}$ m, $R_{max} = 1.99 \cdot 10^{-3}$ m, and correspond to values of 9.7 and 16.4 σ_x for the minimum and maximum HEL radii respectively.

In order to correct this, values of $n = 1.65$ and $g = 17$ were used in the SixTrack collimator input file in order to obtain the correct values for the HEL radii at the elens routine. It is not understood why this was necessary, however the author does not have the knowledge or experience of SixTrack and its collimation routine to propose an explanation. In the near future the SixTrack HEL process will be updated by Miriam Fitterer (Fermilab) and Joschka Wagner (CERN), and may include the novel models from the MERLIN HEL process.

Figure 5.55 shows the ensuing discrepancy when using the initial input values ($n = 4$, and $g = 1.7$), compared to the adjusted values ($n = 1.65$ and $g = 17$). It is clear that the DC HEL effect begins at around $1.17 \cdot 10^{-3}$ m, as expected, whereas the initial value Poincaré section is akin to that produced with no HEL in use. This confirms the R_{min} value of 9.7 σ given when requesting a value of 4 σ in the collimator input file.

After confirming that the elens process in SixTrack is acting on the correct part of the proton bunch, we may observe the expected effect of the different HEL operation modes on the beam halo. In order to benchmark the MERLIN HEL process we may compare these modes using the Poincaré sections.

5.6.1.2 HEL Profile

In both SixTrack and MERLIN, there exist two HEL profiles. These are the selection functions $f(r)$ used to calculate the kick θ_{kick} shown in equation 3.7 as opposed to the maximum kick θ_{max} shown in equation 3.5. The 'radial' profile is a parameterisation of the measured profile of the prototype LHC HEL cathode. The selection function, shown in equation 3.6, is referred to as the 'perfect' or 'simple' profile; the profile given for a perfectly symmetrical e^- distribution with uniform charge density.

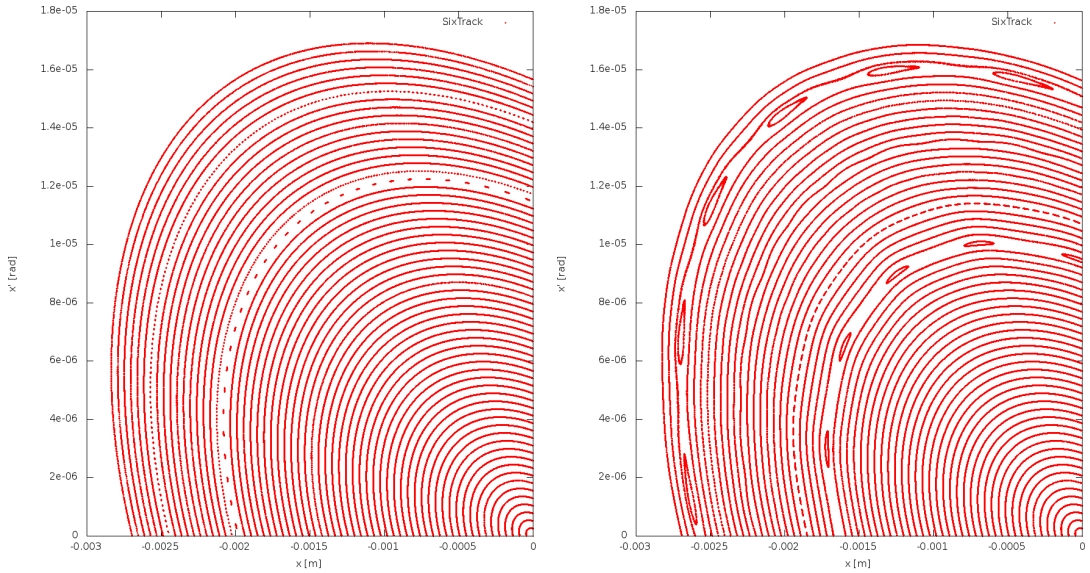

Figure 5.55: Poincaré section for a DC 'Tevatron' HEL in the nominal LHC from SixTrack simulations, comparing the original (left), and adjusted (right) input R_{min}. Particles are initially populated between 0 - 10 σ. We can see without the adjustment that there is no influence from the HEL, and after making the adjustment the islands shown in the right plot show the expected HEL behaviour.

Figure 5.56 compares the two profiles for the Tevatron HEL hardware in the nominal LHC, as expected both codes are equivalent.

It is important to note that, as mentioned previously, the HEL force is attractive to protons, thus the profiles in Fig. 5.56 are in fact only magnitudes as the actual kick is negative, i.e. towards the centre of the beam axis. Also all models are constructed such that if a particle is within the HEL minimum radius R_{min}, there is no kick, a result of the assumption of uniform electron density and radial symmetry of the HEL beam. Also the HEL will effect particles outside the maximum HEL radius R_{max}, the kick being similar to that from a line charge at the centre axis.

We note that the radial profile maximum is not at R_{max} as we would expect. This could be due an artifact from measurement, or some other error. We will assume that this is correct however in order to compare directly with SixTrack.

5.6.1.3 Poincaré Section Comparisons

It is important to note that there are differences in the tracking between MERLIN and SixTrack. We use the thick lens symplectic tracker in MERLIN, whereas a thin lens

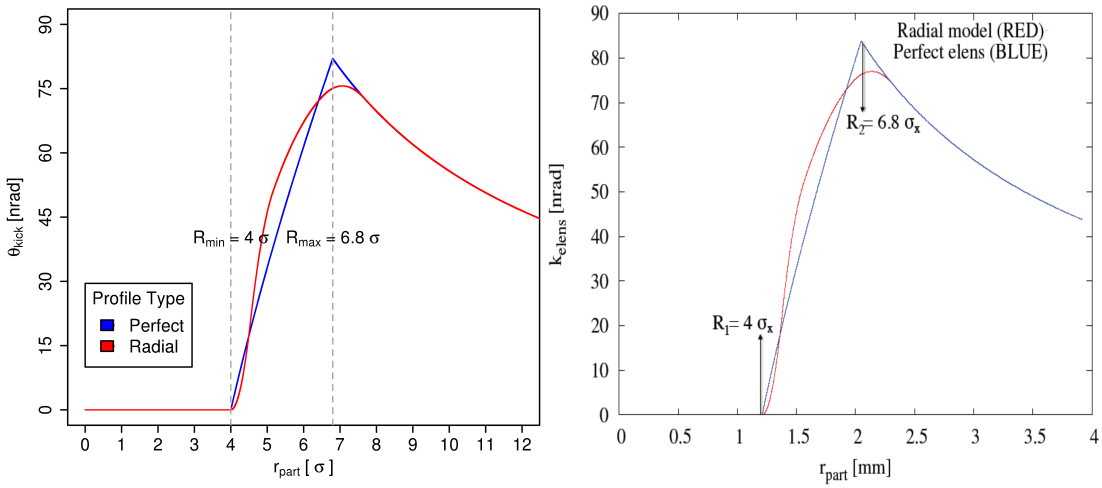

Figure 5.56: HEL radial kick profiles, comparing `MERLIN` (right) with `SixTrack` (left), for identical Tevatron HEL parameters. As expected the profiles are identical in MERLIN and SixTrack.

symplectic tracker is used in SixTrack. The difference is evident from Fig. 5.57, where the Poincaré sections for 64 equally space protons between 0 - 10 σ_x are plotted for 10^4 turns in the nominal LHC, at the position of the HEL. This initial distribution sets all other initial particle coordinates to 0, which means that it represents an ideal bunch with no transverse momentum components, and no longitudinal displacement. This is not representative of an accelerator beam, and is only used to demonstrate the effect of the HEL on what should otherwise be perfectly stable motion in the accelerator, which is represented as smooth ellipses in phase space as shown in Fig. 5.57.

Due to the differences in tracking between MERLIN and SixTrack we do not expect identical results when plotting the Poincaré sections for HEL operation modes. The first and simplest operation mode to be implemented is the DC mode, in which the HEL constantly runs at maximum current, and collimation enhancement relies on the coupling with machine resonances. This is manifest in the formation of islands in the Poincaré section as shown in Fig. 5.58.

We observe in Fig. 5.58, that in both MERLIN and SixTrack there is no perturbation to normal particle motion inside the HEL minimum radius R_{min}, this is as expected as both models of the HEL assume no effect inside R_{min}. In SixTrack islands are created near 4 σ_x, whereas in MERLIN perturbations take the form of small ripples. It was later found that the machine optics used in MERLIN was different to that in SixTrack. Identical optics were not used because there was no way of reverse engineering the

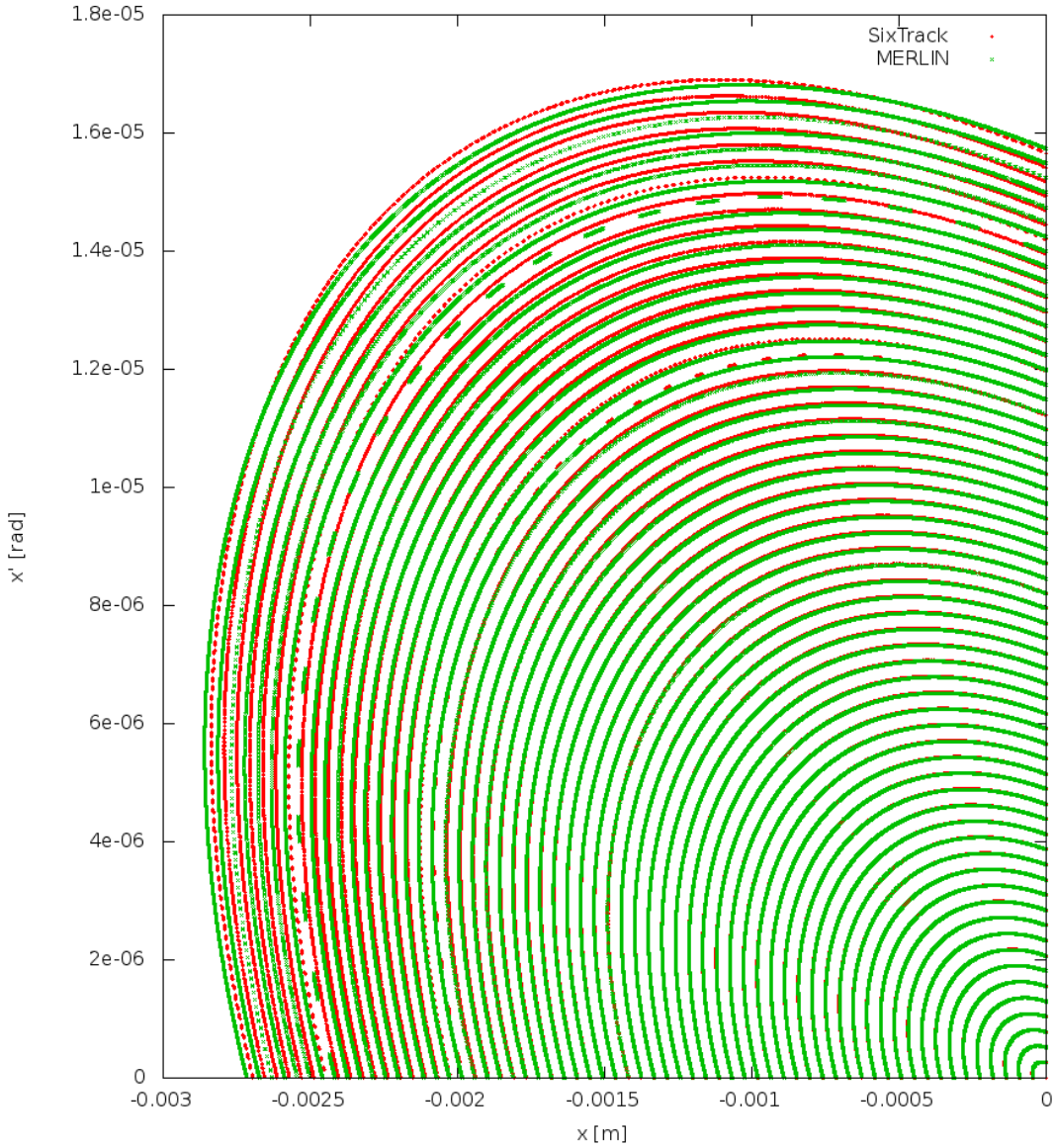

Figure 5.57: Poincaré section for a 64 protons in the nominal LHC, comparing SixTrack (red), and MERLIN (green) tracking. Particles are initially populated between 0 - 10 σ_x. We observe small differences at large displacement due to differences in tracking between MERLIN and SixTrack, this could also be due to slightly different optics settings which were found later.

existing SixTrack generated input files (from the resurrected HEL version of SixTrack) to provide the required MERLIN input. Instead the same lattice version was used with standard settings in MERLIN. When combined with tracking differences, this

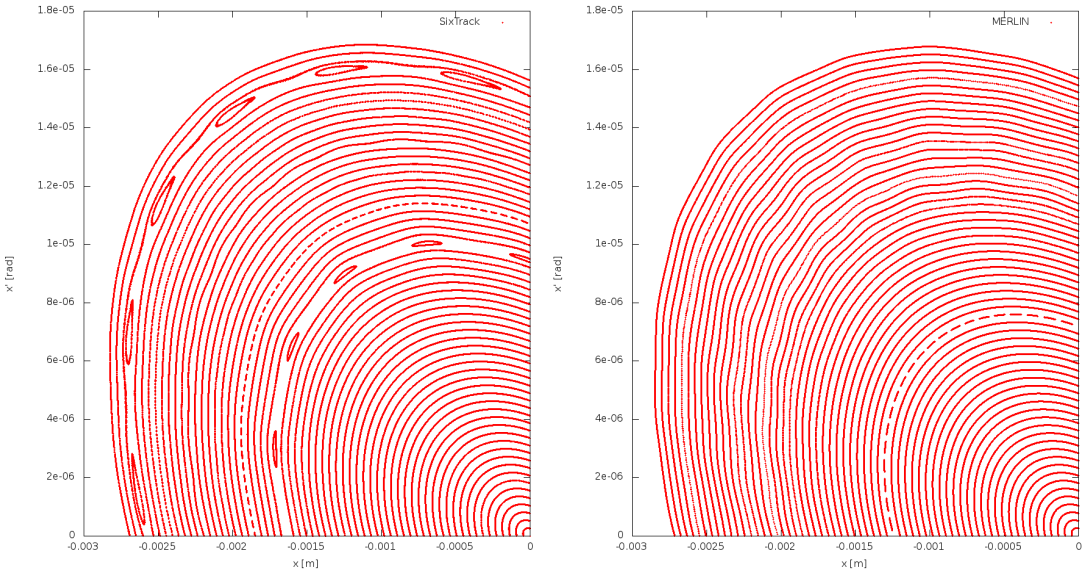

Figure 5.58: Poincaré section for a DC HEL in the nominal LHC, comparing SixTrack (left), and MERLIN (right). Particles are initially populated between 0 - 10 σ_x. The lack of islands in MERLIN simulations could be due to octupole and chromaticity settings differences in the lattice used.

may account for variation between the two codes as the Poincaré sections are affected by octupole and chromaticity settings. Octupoles drive resonances, and chromaticity affects how the particles are swept over these resonances. The perturbations are of similar magnitudes in both codes.

The AC mode relies on driving the betatron oscillations of the protons by modulating the HEL current at a frequency in resonance with the machine tune. This mode takes a large number of variables in order to define the modulation, all of which are described in Chapter 4. A full investigation of this mode, including the optimisation of the parameters used, was performed in [68], and has not been repeated using MERLIN. Instead the optimal parameter settings (referred to as H20 in the previous investigation) from [68] is used for all AC operation in this thesis. The effect of this mode is shown in Fig. 5.59, where as well as the ripples caused by the DC operation, the Poincaré sections are widened in both MERLIN and SixTrack. Tracks from SixTrack are more diffuse than those from MERLIN, there are many possible reasons for this, including the lack of optimisation of the AC parameters in MERLIN, and possible differences in octupole settings.

When considering the HEL as a collimation enhancer, our aim is to force halo

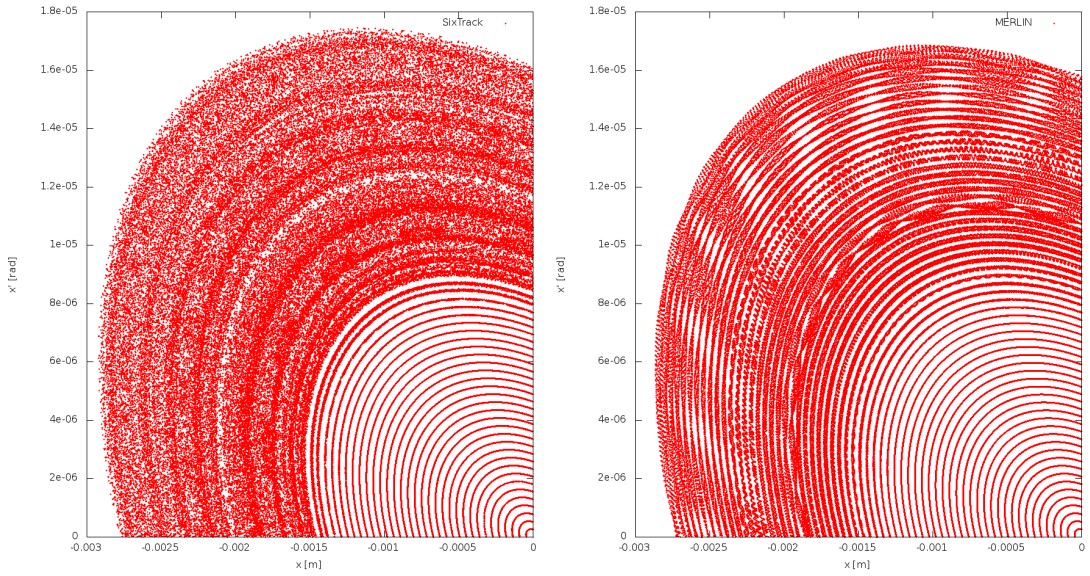

Figure 5.59: Poincaré section for an AC HEL in the nominal LHC, comparing SixTrack (left), and MERLIN (right). Particles are initially populated between 0 - 10 σ_x.

particles onto collimators. The primary collimator is typically located at a position of $\approx 6\sigma$ in the respective plane. Thus our goal is to excite a proton that exists between 4 - 6 σ, to a larger transverse displacement such that it will be absorbed by the collimation system. In the case of a DC HEL, this will only occur if the proton crosses a resonance - accelerators are designed to minimise the probability of this. The proton must also have a transverse displacement near that of the collimator jaw, as the HEL gives only a small displacement in transverse phase space.

With the AC HEL, the widening of particle tracks in transverse phase space is observed to be larger than the displacement given by the DC HEL. This means that collimation enhancement should be greater for the AC mode when compared to the DC mode, as more halo particles will be excited to a displacement large enough to be intercepted by the collimation system.

Diffusive HEL operation gives a random kick to the halo on a turn by turn basis in order to enhance the diffusion of halo particles onto a collimator. Figure 5.60 compares the diffusive HEL operation in MERLIN and SixTrack. As expected the transverse displacement in both codes is much larger than all other operation modes. Not only does this mode offer the greatest collimation enhancement, it is not dependent on rigorous knowledge of the machine tune as in the AC case. Comparing the maximum displacement after 10^4 turns, shown in Fig. 5.60, we may consider MERLIN and SixTrack to

be similar.

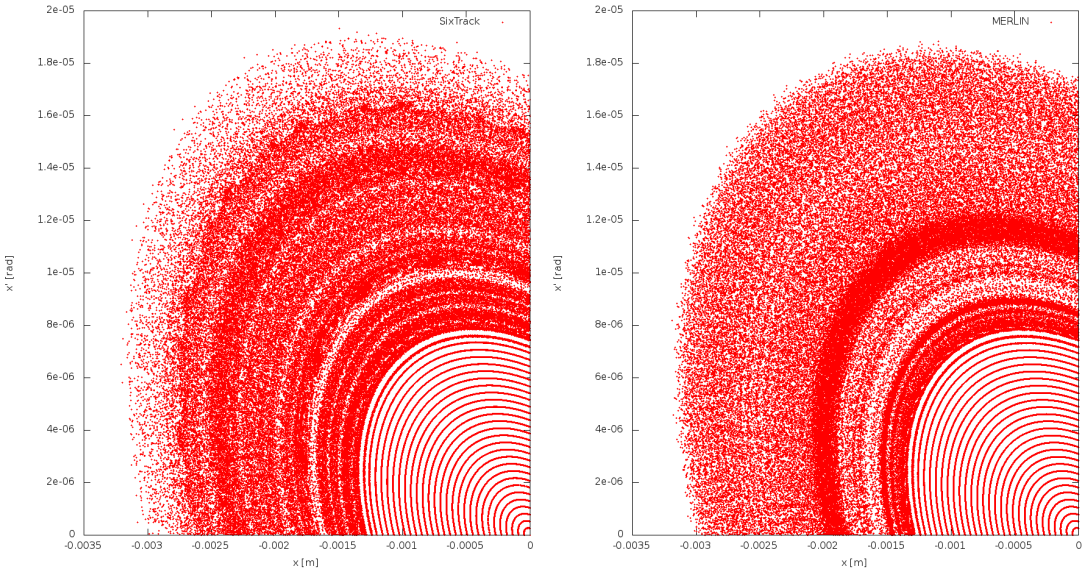

Figure 5.60: Poincaré section for a diffusive HEL in the nominal LHC, comparing SixTrack (left), and MERLIN (right). Particles are initially populated between 0 - 10 σ_x. As the diffusive mode is random it is difficult to compare the codes, however both MERLIN and SixTrack cause similar growth in the Poincaré section.

By directly comparing the DC, AC, and diffusive operation modes of the HEL in MERLIN and SixTrack, we may conclude that SixTracks additional physics processes and thin lens tracking cause only a small difference between the two codes, there is also a small effect from the different optics used. In reality the variation in HEL Poincaré sections due to the differences between MERLIN and SixTrack are small, and for a large number of particles and turns are likely to be negligible.

5.6.1.4 Real Space Footprint

A MERLIN HELHaloDistribution was used to create two LHC bunches of 10^3 protons, one populated between 0 - 4 σ which will be referred to as the core, and the other populated between 4 - 6 σ which will be referred to as the halo. These initial distributions are shown in green in Fig. 5.61, together with their Poincaré sections for 100 turns in purple. The HEL R_{min} and R_{max} are indicated in red.

From Fig. 5.61 we see that the beam is round, which is one of the reasons for the selection of this position in the nominal LHC, we also observe that despite initially populating the core up to a maximum of 4 σ, a very small percentage of particles may

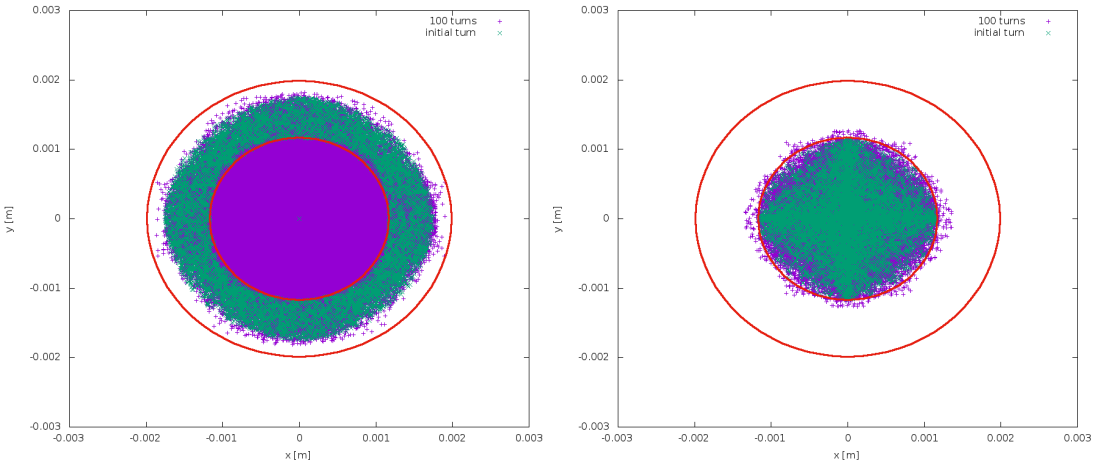

Figure 5.61: HEL footprint at the position of the HEL in the nominal LHC; red lines show R_{min} and R_{max}, green points are the initial distribution, purple are a Poincaré section of this bunch over 100 turns. The left plot shows an initial halo distribution between 4 - 6 σ_x, the right plot shows an initial core distribution between 0 - 4 σ_x.

have a larger transverse displacement for a few turns, and thus interact with the HEL. As the HEL kick is very small, interaction for a small percentage of particles for only a few turns will have a negligible effect, therefore we are not concerned with long term tracking of the core. It is clear that for a halo distribution, particles traverse the area inside 0 - 4 σ despite being initially populated outside this range, thus the HEL does not necessarily operate on a halo particle at every turn.

We must also note that if simulating a halo bunch populated between 4 - 6 σ, in the presence of a collimator with an insertion of 6 σ, a small percentage of the halo will impact upon the collimator without the presence of a HEL. Therefore it is more prudent to simulate a halo between 4 - 5.8 σ so that we may have negligible losses in the case where no HEL is present in the lattice, and thus compare the HEL cleaning enhancement to that of no enhancement when no HEL is present.

5.6.2 Collimation Enhancement Comparison

The initial distribution for HEL simulations is a HELHaloDistribution from $4 - 5.8$ σ with no longitudinal component. Figure 5.62 shows particle survival when using a HEL in the nominal LHC at $s = 10037$ m. Collimator jaw openings are shown in Table 6.1. We observe that the diffusive mode is the only one to enhance collimation in these simulations. MERLIN excels over SixTrack in that it can perform full collimation scattering

and on-line aperture checking with a HEL for many turns within a reasonable simulation time, whereas previous simulations using the HEL in SixTrack omitted scattering in order to minimise run time.

Figure 5.63 shows the effect of doubling the diffusive HEL current; particle survival is roughly halved. The same investigation performed using SixTrack [68] is compared in Fig. 5.63, both codes agree despite the use of full collimation scattering in MERLIN. The impact distribution on the primary collimator is shown for these two currents in Fig. 5.64, we observe that the doubled current increases the likelihood of larger impact parameters.

Insertion Region	**Collimator Family**	**Setting** [σ]
7	Primary	6
	Secondary	7
	Absorber	10
3	Primary	15
	Secondary	18
	Absorber	20
1	Tertiary	8.3
	Absorber	10
5	Tertiary	8.3
	Absorber	10
6	Dump Protection	8
	Secondary	7.5
2	Tertiary	30
8	Tertiary	30

Table 5.9: LHC collimation settings used for HEL simulations.

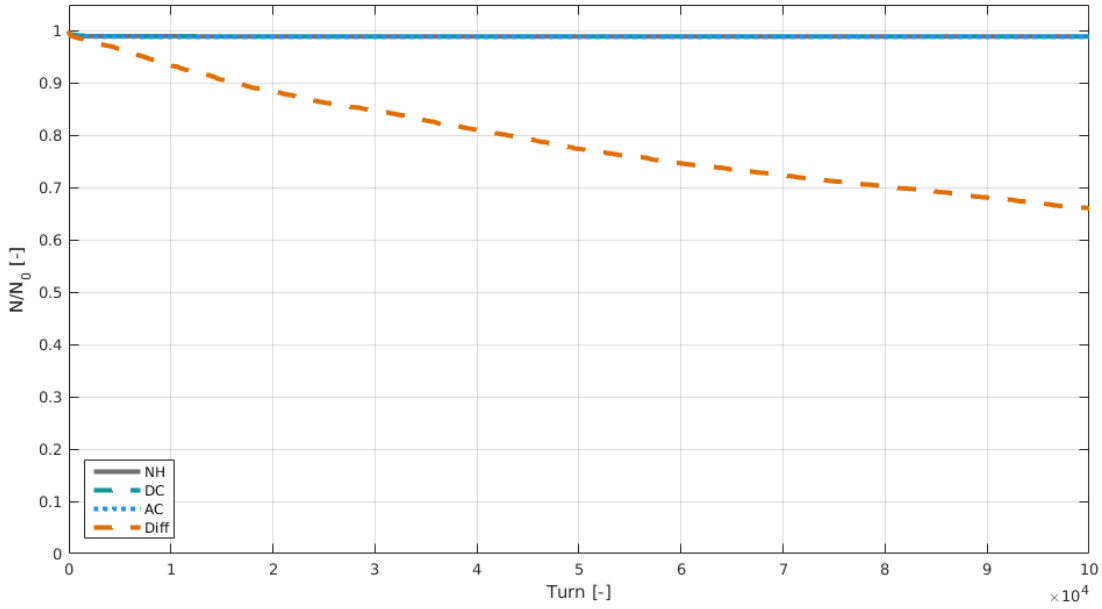

Figure 5.62: Normalised particle survival for 10^5 turns in the nominal LHC, with the HEL using various operation modes, and without a HEL (NH). The initial halo distribution is populated between 4 - 5.8 σ, and the HEL operates between 4 - 6.8 σ. In this simulation full collimation scattering is performed, and only the diffusive HEL operation gives some collimation enhancement.

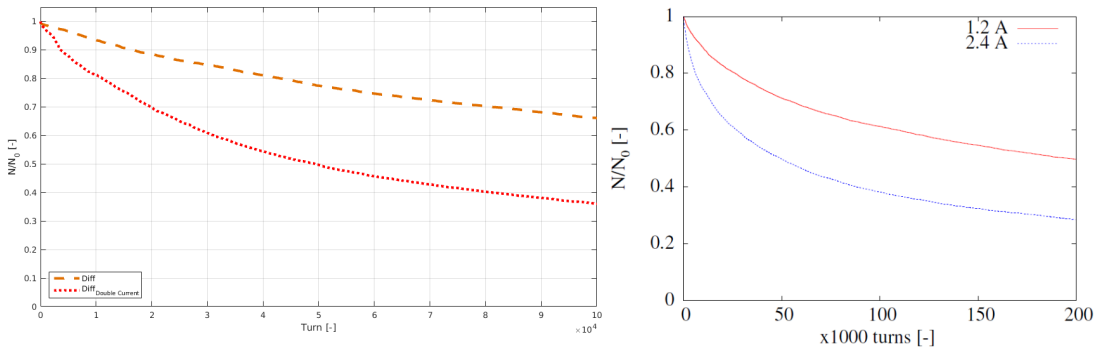

Figure 5.63: Normalised particle survival for 10^5 turns in the nominal LHC, with a diffusive HEL, for an initial halo distribution populated between 4 - 5.8 σ, and a HEL operating between 4 - 6.8 σ (left). Compared to a similar simulation performed with SixTrack (from [68]) for $2 \cdot 10^5$ turns on the right. In both cases two sets of data are shown, one with a diffusive HEL current of 1.2 A, and the second with double the current, 2.4 A. MERLIN and SixTrack are in good agreement, showing similar survival after 10^5 turns.

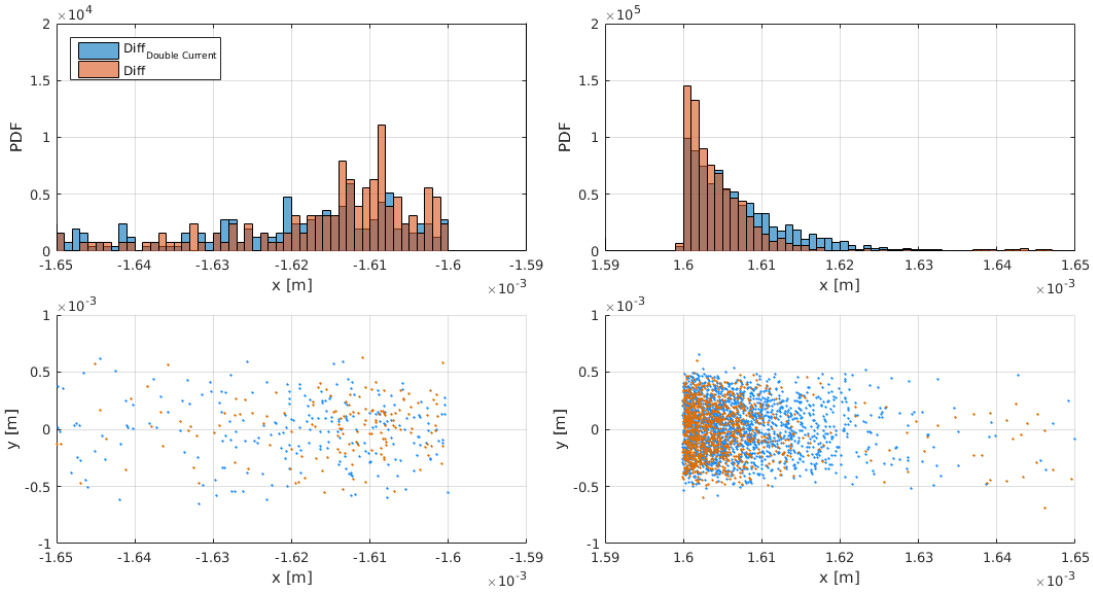

Figure 5.64: The jaw impact distribution for the primary collimator for 10^5 turns in the nominal LHC, with a diffusive HEL, for an initial halo distribution populated between 4 - 5.8 σ, and a HEL operating between 4 - 6.8 σ. The top two plots are histograms of the distributions on the left and right collimator jaws respectively. The bottom two plots show the impacts on the jaw in transverse space. Orange shows data when using a diffusive HEL current of 1.2 A, and the blue with double the current, 2.4 A. Increasing the HEL current increases the impact parameter of protons on the primary collimator.

5.6.3 HL-LHC HEL

For the HL-LHC the HEL is a promising tool for active halo control. The HEL is not currently part of the baseline, this will be decided after operational experience of the LHC at nominal settings (*i.e.* post 2016), when there is better understanding of the collimation system performance at higher energy. In order to inform the decision, we use numerical simulations to estimate the cleaning enhancement of a HEL in the HL-LHC. Inevitably this depends on many factors: the hardware used, as well as the capability of the magnet systems to shape, translate, and maintain the electron beam. The sensitivity of cleaning enhancement on HEL operational parameters (active length, electron current etc.) can be assessed using numerical simulations. First it is important to identify the sensitivity to the position of the HEL in the HL-LHC. We will use beam 1 as the two beams offer similar optics.

The preliminary HEL position in RB46, 88.6 m upstream of IP4, was decided because of the available space, and the fact that there are currently no hardware conflicts at this position. The transverse beam shape in real space (x, y) is not round at this position. As discussed earlier the HEL beam is considered to be perfectly round. Due to the beam shape at IP4 - 88.6 m, this will be referred to as the 'non-round' position.

In order to gauge the sensitivity of HEL cleaning enhancement to the HEL position, two subsequent HEL integration points have been chosen for comparison. The first position offers a round beam ($\beta_x \approx \beta_y$), and is located at ($s = IP4 - 30\ m$). This will be referred to as the 'round' position. The final identified position offers a more elliptical beam than the non-round position, and is located at ($s = IP4 - 119\ m$), this will be referred to as the 'oval' position.

Figure 5.65 shows the beta functions at the three identified positions, from which the difference in beam roundness is clear. The dispersion in this region is low as it is an interaction point, this is shown in Fig. 5.66.

We have benchmarked the HEL process in MERLIN against that in SixTrack, and will use MERLIN to:

1. Compare the effect of HEL operation modes at the round and non-round HEL positions in the HL-LHC to identify any unexpected differences from the LHC behaviour.

2. Confirm that, as expected, the HEL cleaning enhancement is reduced at the non-round position when compared to the round position.

3. Confirm that, as expected, the HEL cleaning enhancement decreases as the beam

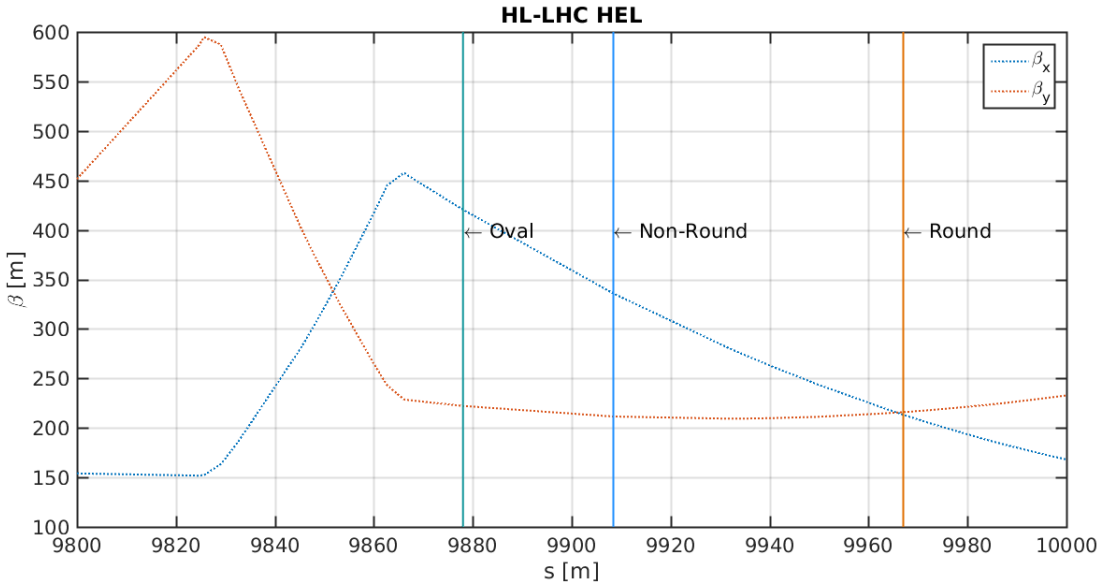

Figure 5.65: Beta functions at the positions of the three identified HEL locations for this study. We see that the beam is round ($\sqrt{\frac{\beta_x}{\beta_y}} \approx 1$) at the round position, and becomes less round at the non-round and oval positions.

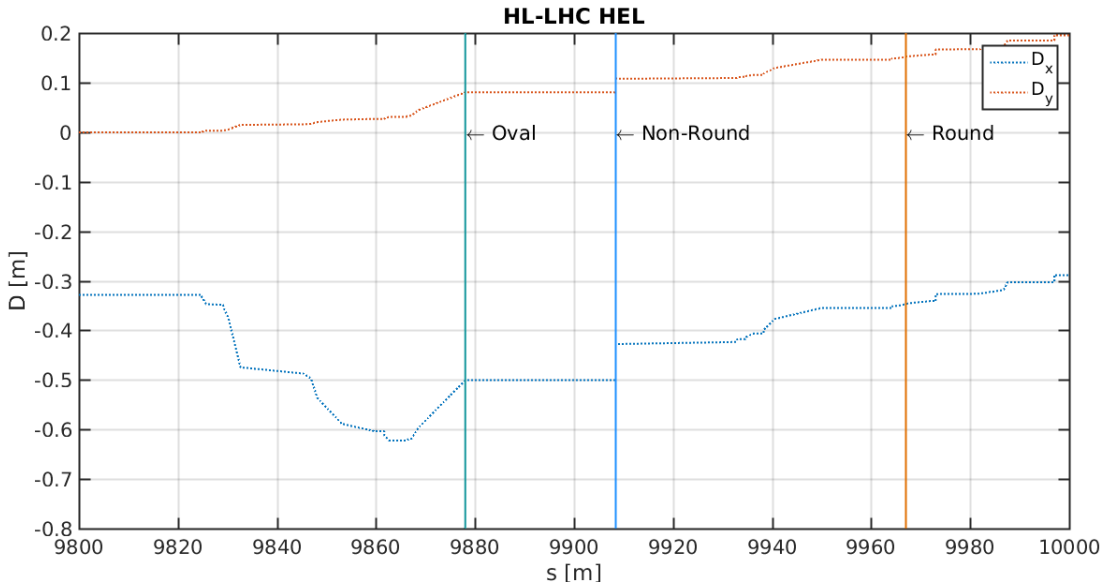

Figure 5.66: Dispersion functions at the positions of the three identified HEL locations for this study. The dispersion is similar at the three positions.

becomes less round (*i.e.* using the oval position).

4. Obtain the most successful existing HEL operation mode at each of the three integration positions.

5. Propose and investigate novel geometrical operation modes to improve the cleaning enhancement at the non-round and oval positions, in order to negate the expected decrease in collimation enhancement at non-round HEL positions.

6. Investigate the sensitivity of cleaning enhancement on HEL hardware parameters, including the HEL current, the HEL beam energy, and the HEL active length.

In following this plan of research not only will we inform the design of the HL-LHC, but we will demonstrate the use of MERLIN as a complete tool for collimation with additional physics processes.

5.6.3.1 HEL Profile

We see from equation 3.5 that the maximum HEL kick is inversely proportional to the HEL maximum radius R_{max}. Thus at positions where the beam size is larger, the maximum HEL kick is smaller. This is shown in Fig. 5.67, where θ_{max} is larger for the round beam due to the smaller beam size in x.

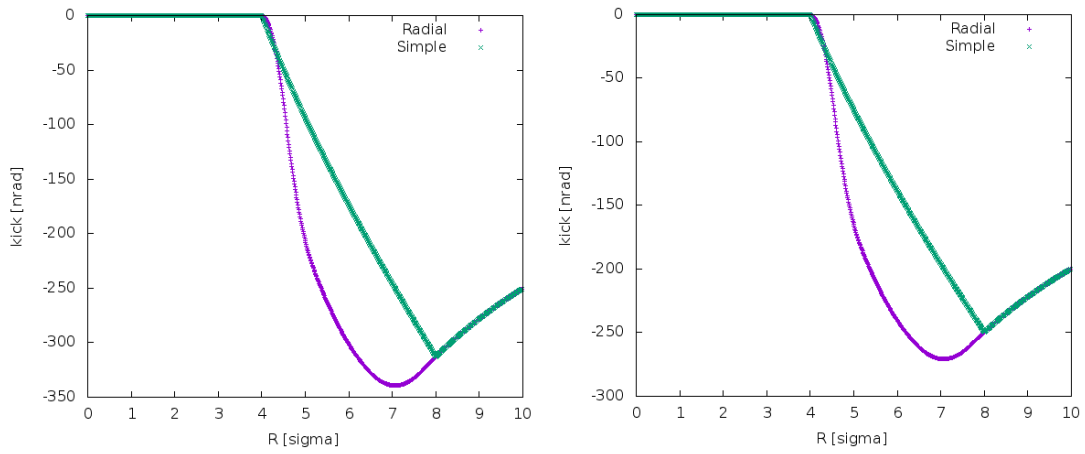

Figure 5.67: HEL radial kick profiles taken along the transverse horizontal axis, comparing the round (left) and non-round (right) positions for the LHC HEL in the HL-LHC, where $R_{min} = 4\ \sigma_x$ and $R_{max} = 8\ \sigma_x$. Both the perfect (simple) and radial models are shown. The radial profiles are not as expected (see Fig. 5.56).

Figure 5.67 also shows us that the radial model is no longer applicable, and gives a θ_{max} larger than the theoretical maximum. The radial profile is a parameterisation of the prototype cathode which had $g = \frac{R_{max}}{R_{min}} = 1.7$, whereas the LHC cathode is defined to have $g = 2$ [70]. By using a radial profile, numerical studies are brought closer to reality, thus the parameterisation for the radial profile was empirically adjusted until the expected profile was obtained (*i.e.* one showing a similar relationship between measured and theoretical profiles as in Fig. 5.56). The result is shown in Fig. 5.68.

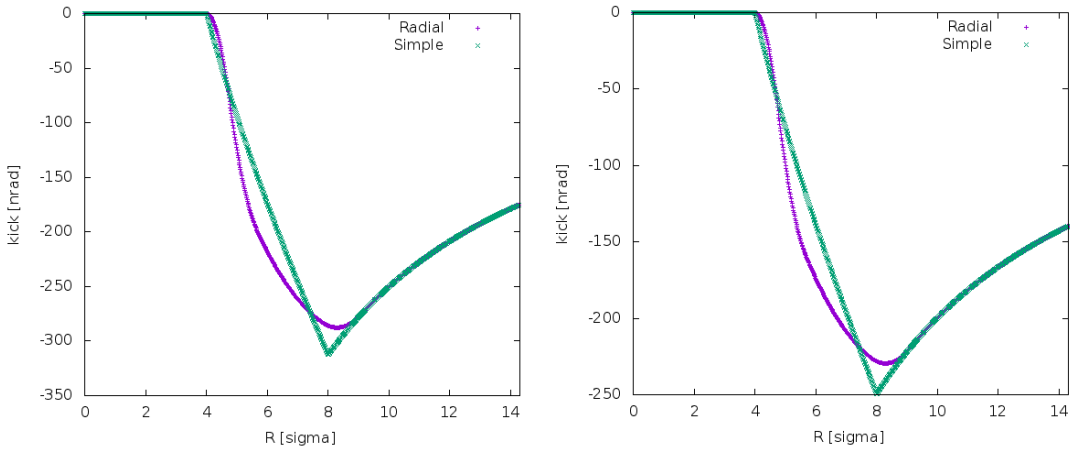

Figure 5.68: HEL radial kick profiles taken along the transverse horizontal axis, comparing the round (left) and non-round (right) positions for the LHC HEL in the HL-LHC, where $R_{min} = 4\ \sigma_x$ and $R_{max} = 8\ \sigma_x$. Both the perfect (simple) and empirically adjusted radial models are shown. The radial profiles are now as expected (similar to those in Fig. 5.56).

The corrected radial profile will be used for all simulations that follow.

5.6.3.2 Real Space Footprint

A MERLIN HELHaloDistribution was used to create two HL-LHC bunches, one for the core populated between 0 - 4 σ, and the other for the halo, populated between 4 - 6 σ. The initial distributions, the footprint of these bunches for 100 turns, and an indication of R_{min} and R_{max} are shown in Fig. 5.69 at the non-round position, and Fig. 5.70 for the round position.

At the non-round position we observe the larger HEL radii, and the fact that R_{min} touches the extremities of the core only in the horizontal. This results in a smaller overlap of the beam halo and the HEL, as shown in the halo footprint (right plot).

At the round position the HEL radii are smaller in real space, and as seen for the

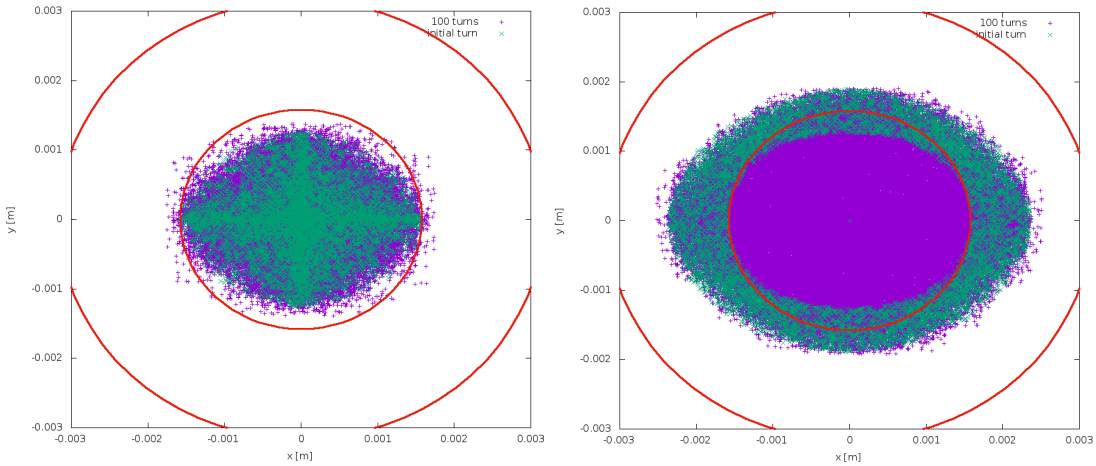

Figure 5.69: HEL footprint at the non-round position of the HEL in the HL-LHC; red lines show R_{min} and R_{max}, green points are the initial distribution, purple are a Poincaré section of this bunch over 100 turns. The left plot shows an initial core distribution between 0 - 4 σ_x, the right plot shows an initial halo distribution between 4 - 6 σ_x. We see that at the non-round position the HEL does not optimally cover the halo as in Fig. 5.70.

nominal LHC case, a small percentage of the core has a transverse displacement larger than 4 σ for a small number of turns. As the beam is nearly round ($\frac{\beta_x}{\beta_y} \approx 1.01$), R_{min} encloses the extremities of the core, meaning a complete overlap with the initial halo. Due to the reduced overlap and thus interaction with the halo at the non-round position, we expect the HEL cleaning enhancement to be reduced when compared to the round position.

In order to improve the cleaning enhancement where the beam is not round, the first novel HEL operation is the elliptical matching proposed in Chapter 3. The resulting footprint of the HEL in the non-round and oval positions are compared in Fig. 5.71. In both cases we choose to match the HEL inner radius to the vertical maximum of the bunch in real space. We see that the HEL radii are larger at the oval position because of the need to avoid any overlap with the beam core.

The second approach to negate the decrease in collimation enhancement at non-round beam positions is the Pogo operation mode. As discussed in Chapter 3 this is possibly more practical than the Hula operation, and may represent more realistic cleaning enhancement in long term numerical simulations. The Pogo operation mode for the non-round and oval positions are compared in Fig. 5.72. In the case we are interested in, *i.e.* where the beam is larger in x, the Pogo operation alternates the elliptical matching between the vertical maxima.

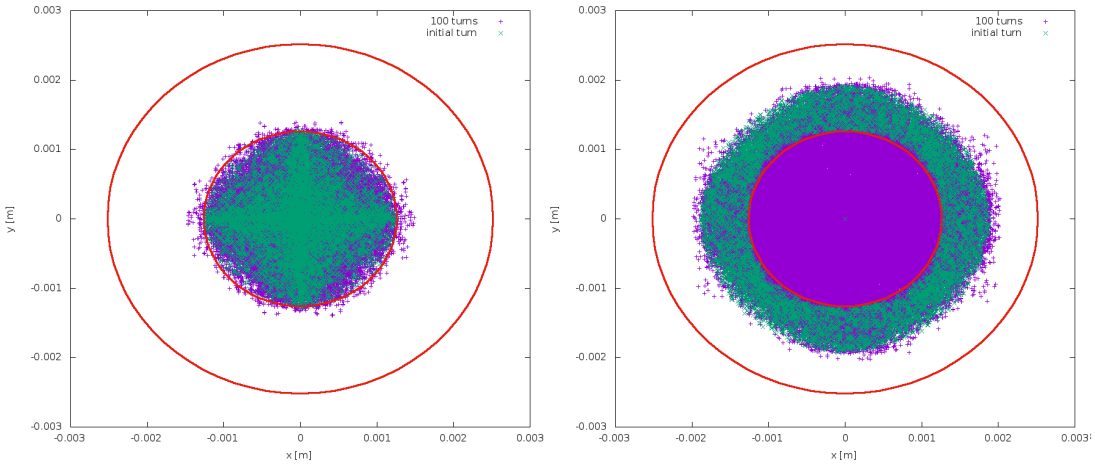

Figure 5.70: HEL footprint at the round position of the HEL in the HL-LHC; red lines show R_{min} and R_{max}, green points are the initial distribution, purple are a Poincaré section of this bunch over 100 turns. The left plot shows an initial core distribution between 0 - 4 σ_x, the right plot shows an initial halo distribution between 4 - 6 σ_x.

The next approach to improving cleaning enhancement at non-round beam positions when using the HEL, is the Hula mode of operation proposed in Chapter 3. In this first dynamic HEL operation mode, we choose to elliptically match the HEL to the vertical maxima, and then translate this HEL around the bunch such that R_{min} touches each horizontal and vertical extremity. This is done in the order shown in Fig. 5.73 in order to imitate a clockwise rotation in real space. As discussed in Chapter 3, this is an approximation as in reality it may not be possible to re-size and translate the HEL beam on a turn by turn basis, it is more likely that many more steps may be required between these maxima, which is likely to reduce the increase in collimation enhancement. Thus we take the Hula operation mode as a best case scenario.

We note that with the Hula operation the HEL no longer covers both sides of the horizontal halo at once, thus decreasing halo coverage. To improve this we use the Close Hula dynamic operation mode. Here the HEL is translated and re-sized each time to allow the smallest possible R_{min} and thus the maximum halo coverage, as shown in Fig. 3.19.

The results of the HL-LHC HEL simulations are presented in Chapter 6, section 6.2.

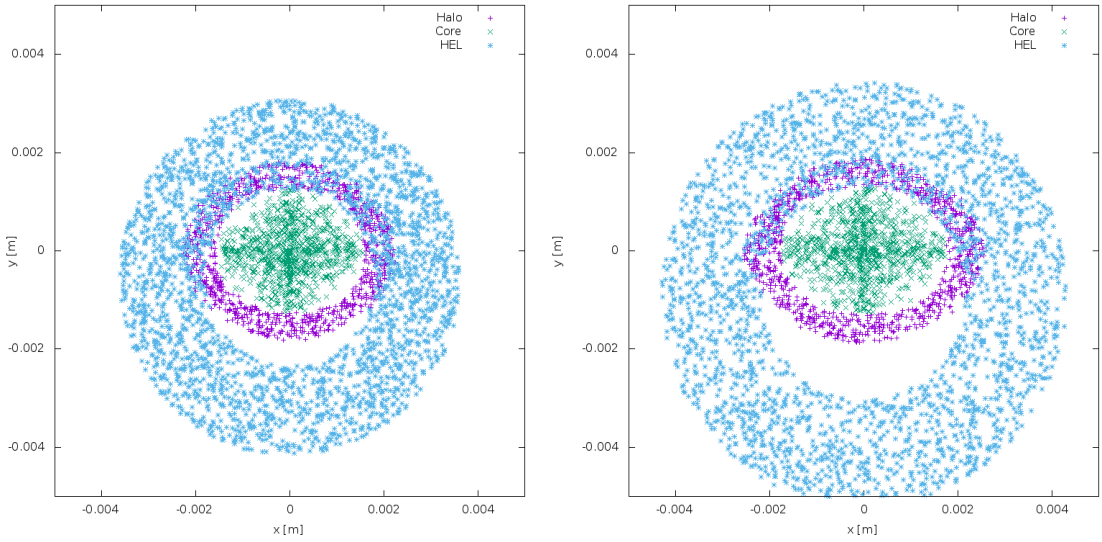

Figure 5.71: HEL footprint for elliptical operation of the HEL in the HL-LHC; blue points indicate the HEL footprint, green points are the core protons, purple are the halo protons. The left plot shows the non-round position, the right plot shows the oval position.

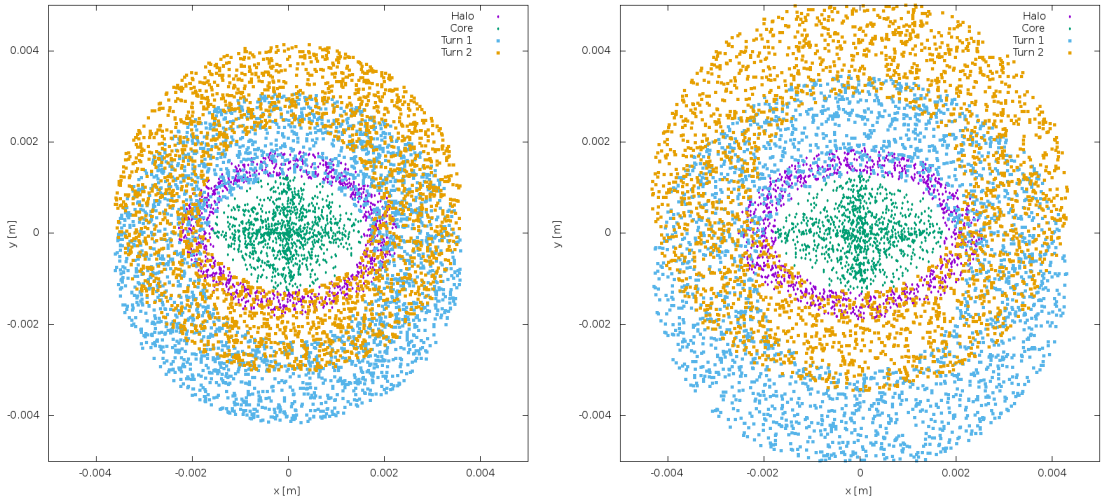

Figure 5.72: HEL footprint for Pogo operation of the HEL in HL-LHC; blue points indicate the HEL footprint odd turns, yellow points indicate the HEL footprint on even turns, green points are the core protons, purple are the halo protons. The left plot shows the non-round position, the right plot shows the oval position.

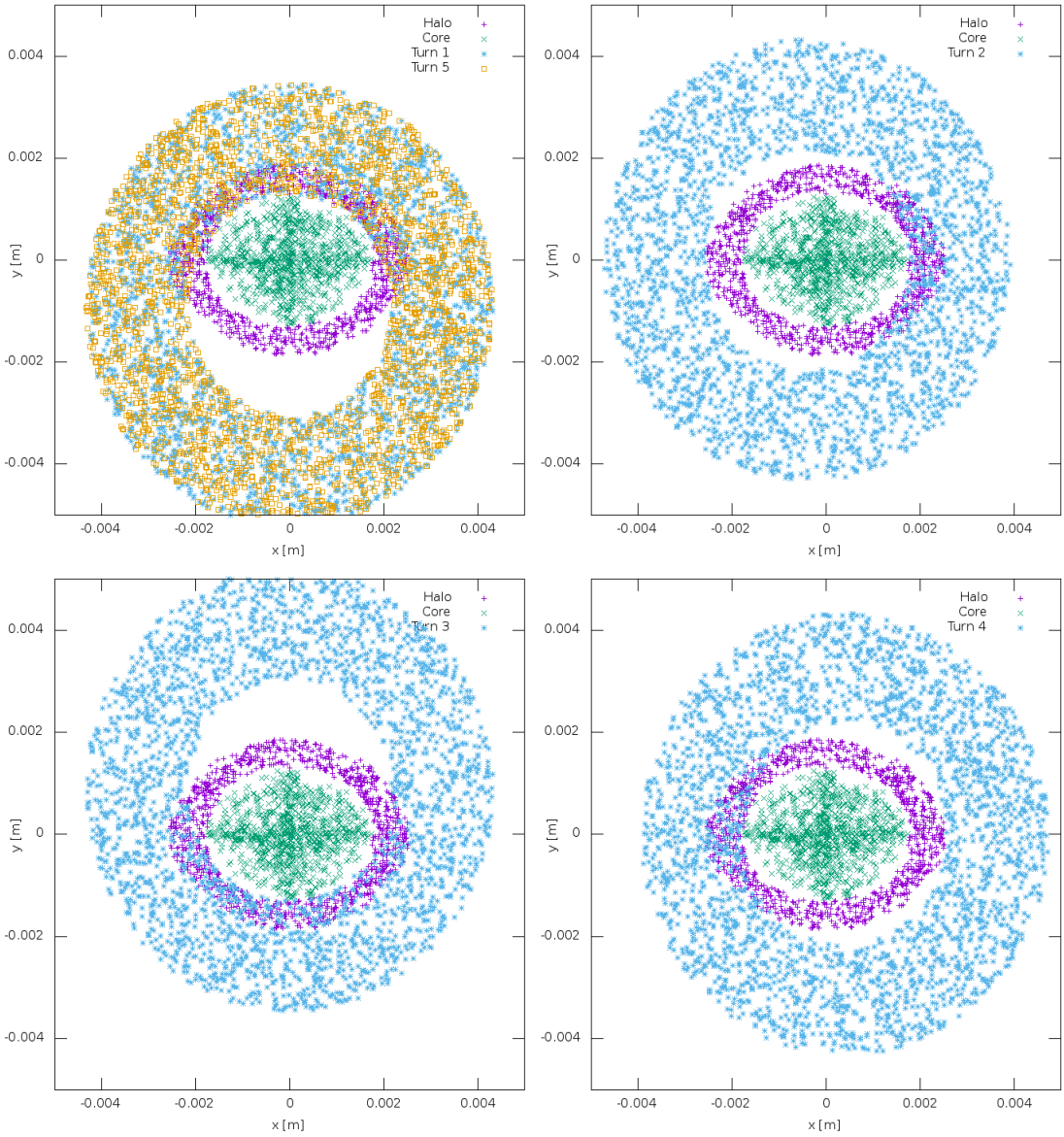

Figure 5.73: HEL footprint for Hula operation of the oval HEL in the HL-LHC; blue and yellow points indicate the HEL footprint, green points are the core protons, purple are the halo protons. The top left plot shows the first and fifth turn, the top right plot shows the second turn, the bottom left plot shows the third turn, and the bottom right plot shows the fourth turn.

6

Results

In this chapter we present the results of novel simulations of the LHC and HL-LHC collimation system, including the effect of dispersion suppressor collimators and novel composite collimator jaw materials on loss maps, and an integration study of the HEL in the HL-LHC.

6.1 Collimation

6.1.1 Nominal LHC

The 7 TeV beam 1 LHC is used to confirm the effect of composite materials over pure elements when simulating collimator jaws, then to produce loss maps with novel collimator materials.

Table 6.1 shows the collimator settings used for the nominal LHC collimation simulations. The initial distribution used is similar to that in the 6.5 TeV beam 2 simulation, without any longitudinal component. A halo distribution in MERLIN is used with an impact parameter of $1\mu m$ on the primary collimator, a 2σ Gaussian in yy' phase space is used. 6D symplectic tracking is used in all of the following simulations unless stated otherwise.

Insertion Region	Collimator Family	Setting [σ]
7	Primary	6
	Secondary	7
	Absorber	10
3	Primary	15
	Secondary	18
	Absorber	20
1	Tertiary	8.3
	Absorber	10
5	Tertiary	8.3
	Absorber	15
6	Dump Protection	8
	Secondary	7.5
2	Tertiary	25
	Injection	999
8	Tertiary	25

Table 6.1: LHC collimator settings used for the nominal 7 TeV LHC beam 1 simulations in SixTrack and MERLIN.

6.1.1.1 Effect of Composite Materials

Loss map simulations were run using pure elements as collimator jaw materials (carbon, copper, tungsten), and compared to those run using composite materials (CFC, Glidcop, Inermet). The full 7 TeV lattice is shown in Fig. 6.1, and IR7 in Fig. 6.2. Table 6.2 shows the differences in losses in IR7, and that composite materials give slightly fewer losses over the entire LHC. As in the 6.5 TeV case, the composite simulation gives around 5% fewer losses in the primaries, 15% more secondary losses, and around twice the number of absorber losses, in IR7. The use of composites results in an increase in cold losses throughout the machine, and a small but important redistribution of losses. As loss location and magnitude prediction in the real machine is our goal, these significant differences show the necessity of accurately modelling composite materials.

Fewer interactions in the primary collimators when using CFC AC150K (as seen in the 0.5 m test case in Chapter 5) leads to more protons escaping and forming the secondary halo. This in turn gives rise to an increase in secondary losses. In the case of absorber losses we have observed in Chapter 5 (Table 5.3) that the composite implementation of Inermet appears to give a slightly larger inelastic contribution to the total cross section. Together with the increase in the secondary halo that escapes from the primary collimator, this is the reason for the rise in absorber losses. The increased DS losses are due to the increase in the single diffractive contribution to the total cross section in the primary collimators given when using CFC AC150K rather than pure carbon, together with the increase in particles escaping the primary collimators. Thus we confirm what was identified in the 6.5 TeV loss map validation simulations.

Figure 6.3 shows the distribution of protons impacting upon the most loaded secondary collimator, comparing the pure and composite material cases. When using pure materials the losses are slightly more concentrated on the outer (negative) jaw. This indicates that large momentum transfer interactions are more frequent in CFC AC150K than pure carbon. This has already been identified in the previous chapter. As Fig. 6.3 shows the probability density function it is not shown that there are around 20% more impacts when using composite materials, as more particles escape the primary collimators.

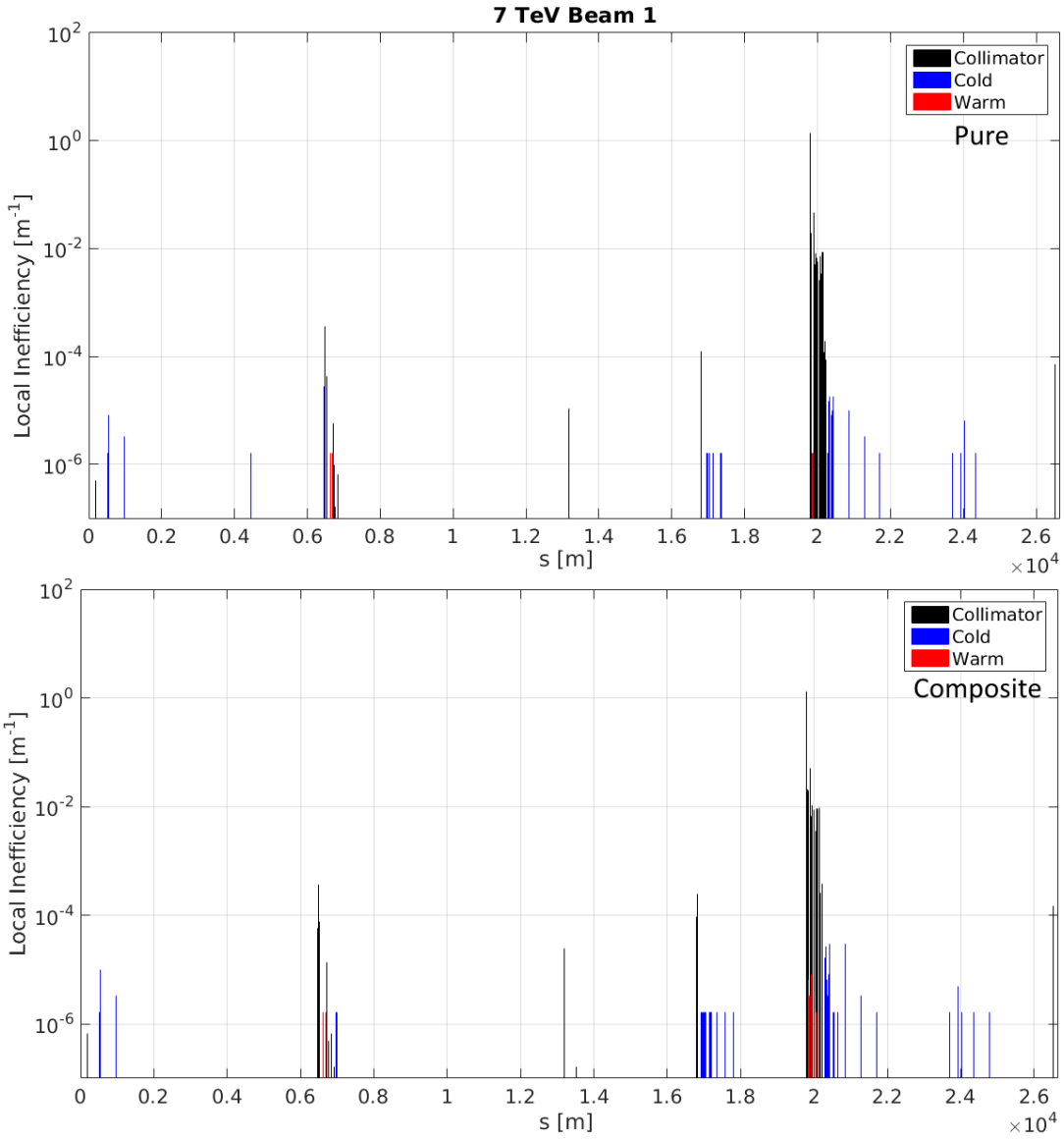

Figure 6.1: Loss maps for beam 1 of the nominal LHC comparing the use of pure elements for collimator jaws (above) with composite materials (below). Using composites gives rise to an overall decrease in collimator losses and a small increase in cold losses. This shows the importance of modelling materials as composites.

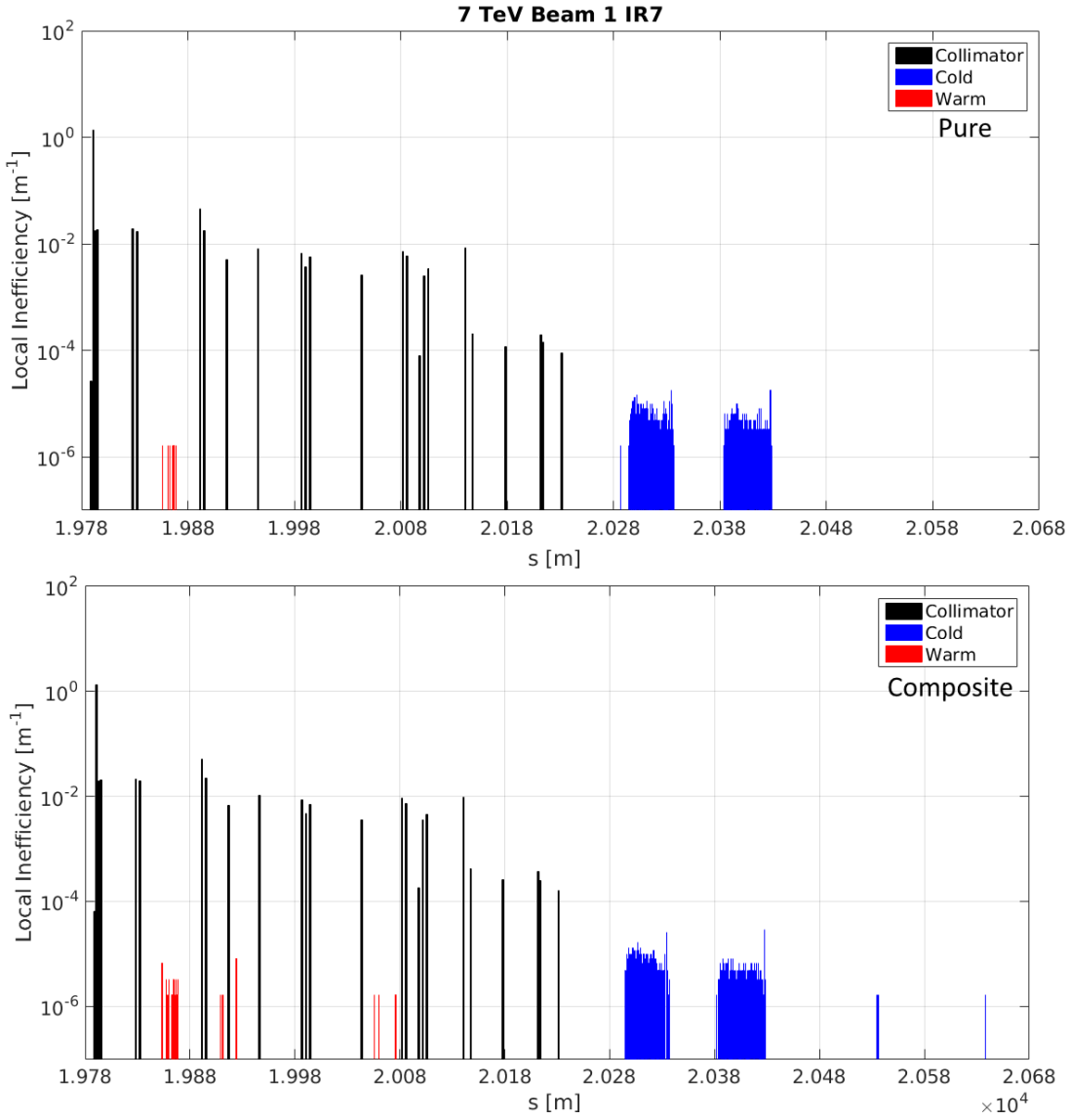

Figure 6.2: Loss map for beam 1 of the nominal LHC comparing the use of pure elements for collimator jaws (above) with composite materials (below), showing losses in IR7. When using composites more protons escape the primary collimators. This gives rise to more secondary, absorber, and cold losses.

		Pure	Composite
Entire LHC	**Collimator**	6116063	6076341
	Cold	1525	1950
	Warm	16	61
IR7	**Collimator**	6111836	6069342
	(Primary)	5135092	4908907
	(Secondary)	972142	1151514
	(Absorber)	4602	8921
	Cold	1404	1766
	Warm	14	57
IR3	**Collimator**	1649	1955
	Cold	32	58
	Warm	2	4

Table 6.2: Breakdown of losses in loss maps for beam 1 of the 7 TeV LHC, comparing the use of pure elements and composites for collimator jaw materials.

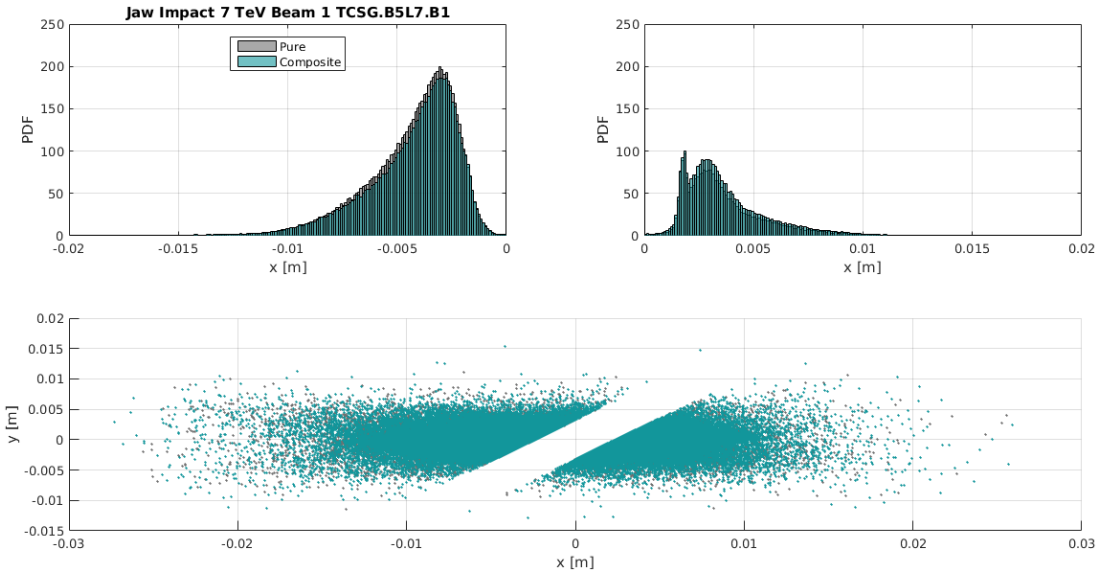

Figure 6.3: Particle impact distribution in the transverse plane for the most loaded secondary collimator TCSG.B5L7.B1, comparing the use of pure and composite materials for secondary collimator jaws. The distribution of losses is changed when using composites, again highlighting the need for proper treatment of materials as this can greatly effect loss location prediction.

6.1.1.2 Effect of Novel Materials

We replace primary collimators with those made of novel materials in order to observe the effect on loss maps. The more likely scenario is that the secondary collimators in the LHC will be replaced for HL-LHC.

Figure 6.4 shows the loss map for MoGr primary collimators, and compares to a similar simulation run with SixTrack. We note that the collimator settings in SixTrack are slightly different, as the tertiary collimators in IR2 and IR8 are inserted, whereas they are more relaxed in the MERLIN simulation. IR7 is compared in Fig. 6.5 for the same simulations, it is clear that there are more DS losses in MERLIN, as well as extra collimators (one primary and one secondary) which is likely due to the use of a newer lattice in MERLIN simulations. Other than these missing collimators and the difference in cold DS losses, the loss patterns appear similar. The use of CuCD primary collimators is shown and compared with SixTrack in Fig. 6.6, with IR7 shown in Fig. 6.7. In both MERLIN and SixTrack there is a significant reduction in the DS losses in IR7 when using CuCD rather than MoGr as the primary collimator material.

As the collimator settings are different in SixTrack these loss maps are only compared to show the agreement in the reduction of IR7 DS losses when using novel primary collimator materials. We may quantitatively compare the difference in losses using Table 6.3, which details the breakdown of important loss locations and magnitude when using novel composite materials for primary or secondary collimators in MERLIN. Table 6.4 displays this information in IR7 as a percentage of losses obtained with standard composite collimators. When replacing the primary collimators with those made of novel composites, there is a 1% increase in primary losses, and a reduction in secondary losses. MoGr gives a 22% decrease in the IR7 DS losses, however CuCD gives a much larger decrease at 48% when compared to standard collimator materials. This indicates the benefit of using these novel materials as they offer an improvement in proton absorption.

Due to the requirements of robustness, it is more likely that secondary collimators will be replaced with novel composite materials. Loss maps generated from simulations with novel secondary collimator materials are compared with those using standard composites. The full loss map when using CuCD is shown in Fig. 6.8, from which we observe a reduction in most losses outside of IR7. Looking closely at IR7 in Fig. 6.9 we remark that primary losses are identical as expected, with a small but noticeable decrease in some secondary losses, and the absorbers. Cold DS losses in IR7 appear to be slightly reduced, Table 6.4 shows that there is a 9% decrease. Though there are

few losses in IR3, fewer losses are recorded when using CuCD secondary collimators as shown in Table 6.3.

The loss map generated with MoGr secondary collimators is shown in Fig. 6.10, the reduction in losses post-IR7 is not so obvious. IR7 is shown in Fig. 6.11, from which we observe identical primary losses, similar secondary losses, and a perceivable difference in absorber losses. There appears to be no reduction in DS losses, which is confirmed in Table 6.4 which shows a 1% increase. It is interesting to note that IR3 losses, shown in Table 6.3, are almost identical, implying that MoGr is much less effective than CuCD in reducing the halo escaping from IR7. Replacing secondary collimator jaws with novel composite materials thus offers comparable cleaning to existing materials, whilst offering other advantages not observable from loss maps, such as increased robustness and reduced impedence.

Figure 6.12 shows the impact distribution of protons on the most loaded secondary collimator, comparing the use of CuCD and MoGr as primary collimator materials. Not shown is the fact that there are $\approx$ 30% fewer impacts from CuCD primaries, a result of more inelastic interactions in CuCD, summarised in Table 6.3. It is more probable for a secondary halo particle from MoGr primaries to impact upon the positive jaw, a consequence of reduced rigidity due to large momentum transfer scatters, in this case due to the slightly larger contribution of single diffractive interactions. The dp distribution of particles impacting upon the most loaded secondary collimator is shown in Fig. 6.13. It is evident that the probability distributions are identical for all primary collimator materials, with the 0.12 GeV cut-off a hallmark of the advanced single diffractive algorithm, showing that this collimator is dominated by particles that have undergone a single diffractive interaction in the primaries.

The loss distribution in the most loaded secondary collimator is shown in Fig. 6.14, from which we observe a hierarchy dominated by the mean free path of each material. As CuCD has a mean free path almost half that of MoGr, there is an increased likelihood of inelastic interactions occurring at the start of the collimator.

It is evident that using novel materials in primary collimators provides the best improvement to DS losses, a reduction of 40% is obtained with CuCD which appears more promising than MoGr.

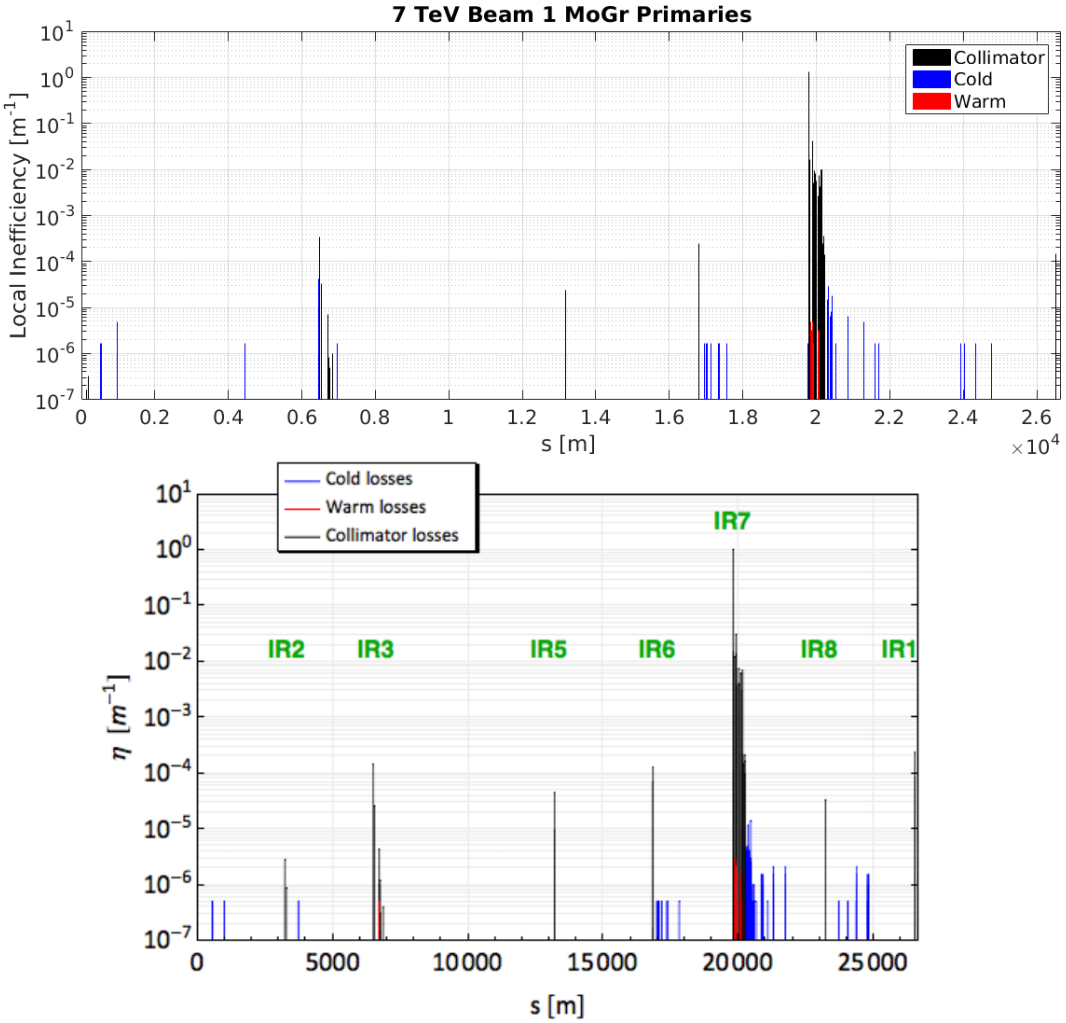

Figure 6.4: Loss map for beam 1 of the nominal LHC, comparing MERLIN (above) with SixTrack [93] (below) using MoGr for primary collimator jaws. As there are more primary and secondary collimators present in the MERLIN simulation (due to using a newer lattice), fewer protons escape IR7. In the SixTrack simulations the tertiary collimators in IP2 and IP8 are inserted, which explains the collimator losses around s = 23000 and s = 3000 m.

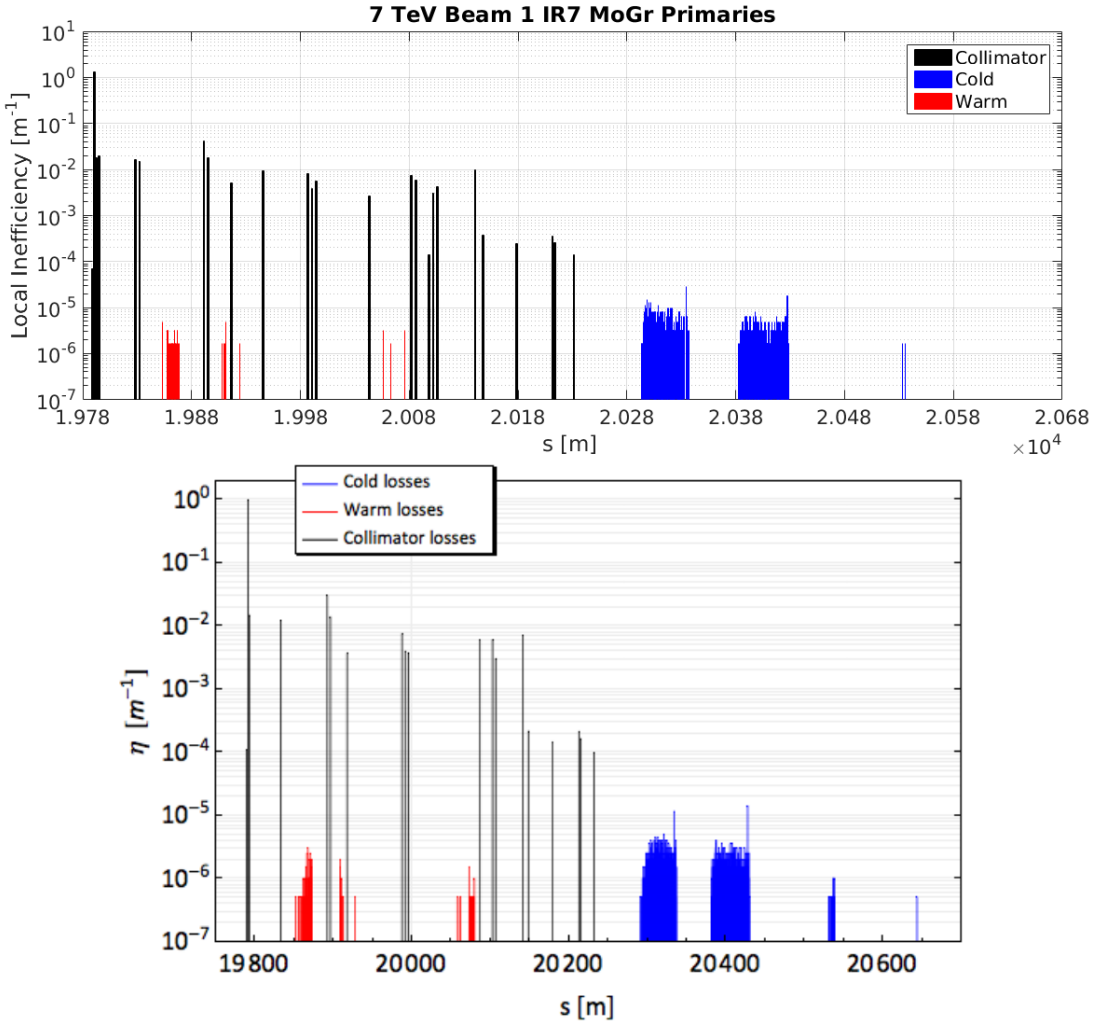

Figure 6.5: Loss map for beam 1 of the nominal LHC in IR7, comparing MERLIN (above) with SixTrack [93] (below) using MoGr for primary collimator jaws. The extra collimators in the MERLIN simulation are evident.

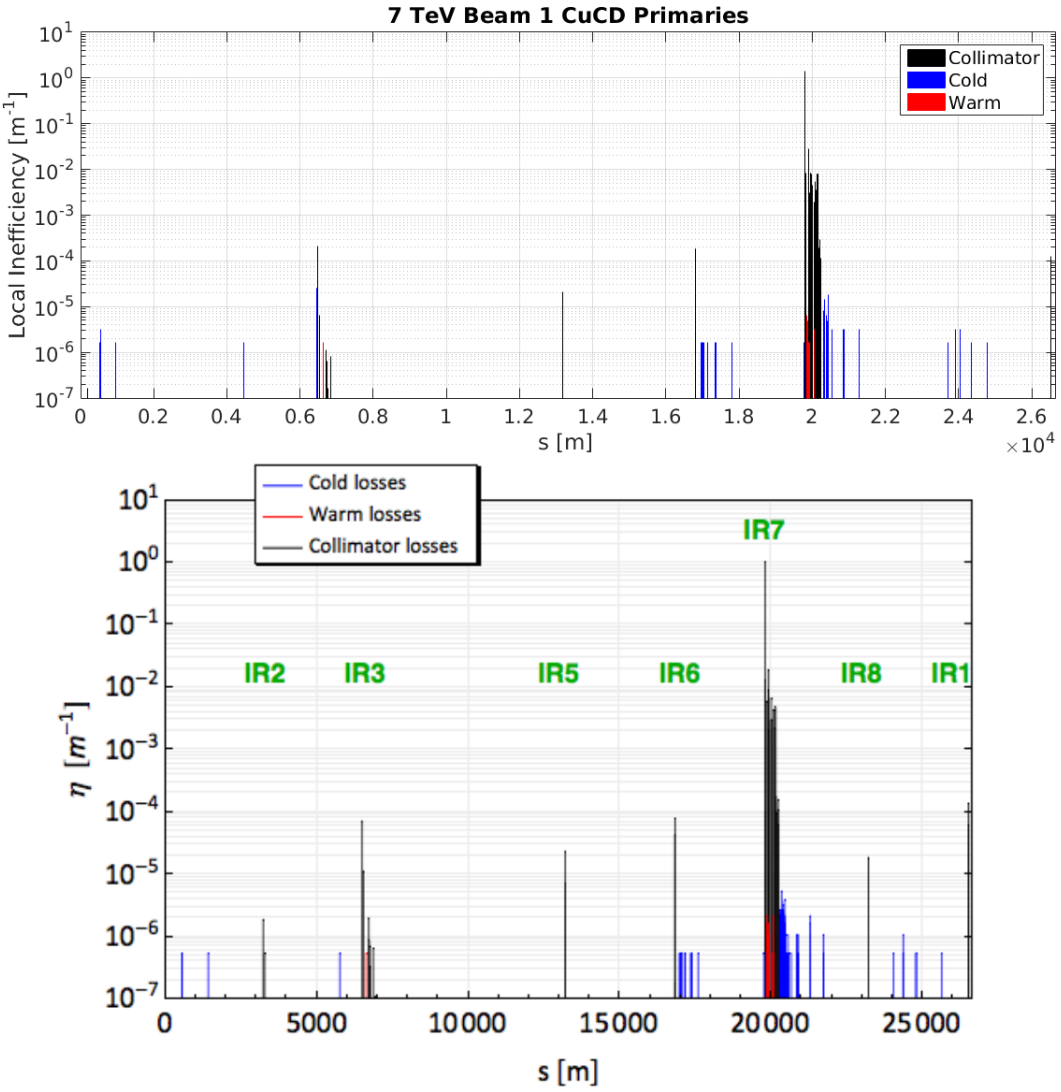

Figure 6.6: Loss map for beam 1 of the nominal LHC, comparing MERLIN (above) with SixTrack [93] (below) using CuCD for primary collimator jaws.

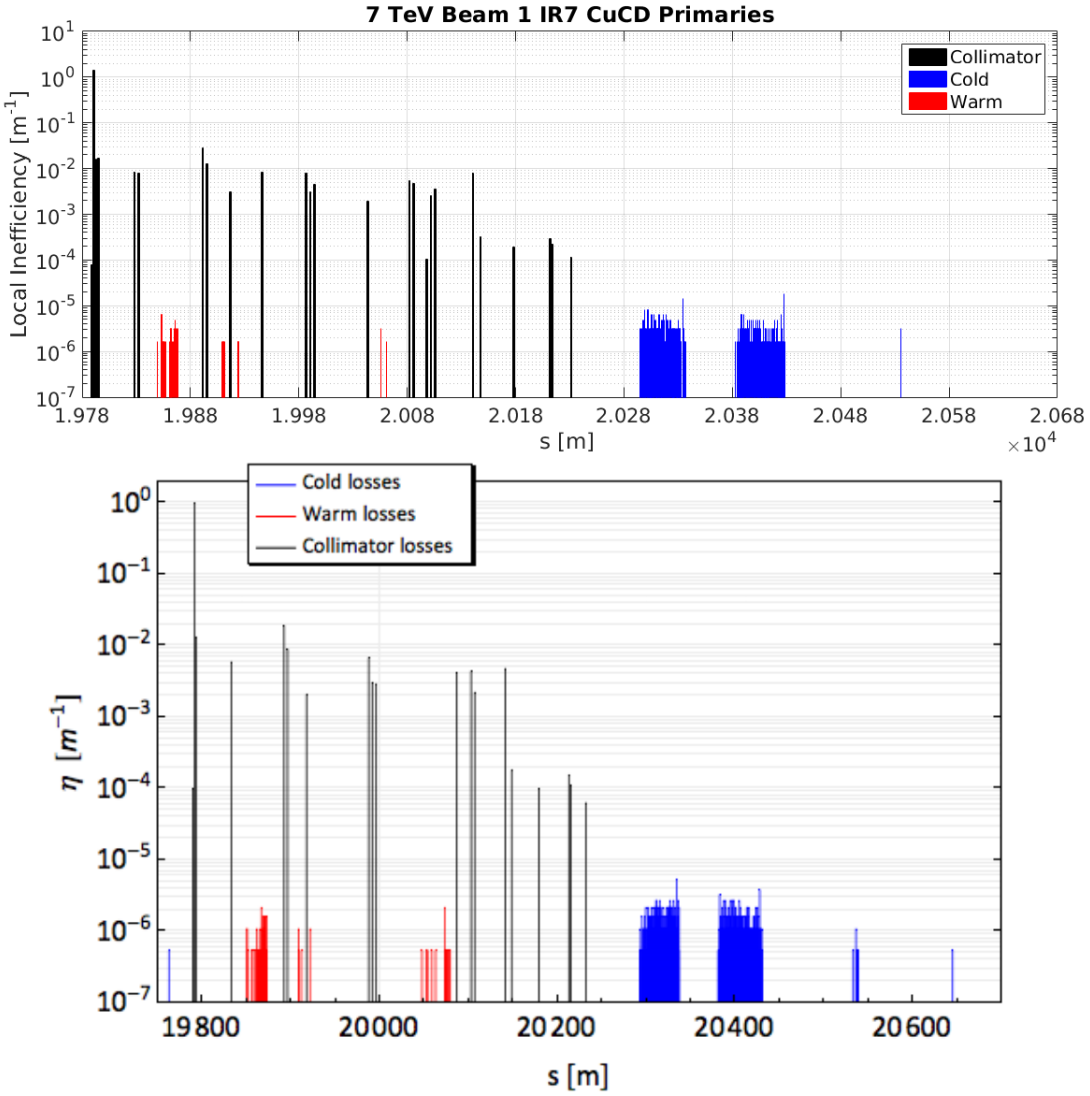

Figure 6.7: Loss map for beam 1 of the nominal LHC in IR7, comparing MERLIN (above) with SixTrack [93] (below) using CuCD for primary collimator jaws. In both codes we observe a significant decrease in cold losses in IR7 as compared to using MoGr primary collimators.

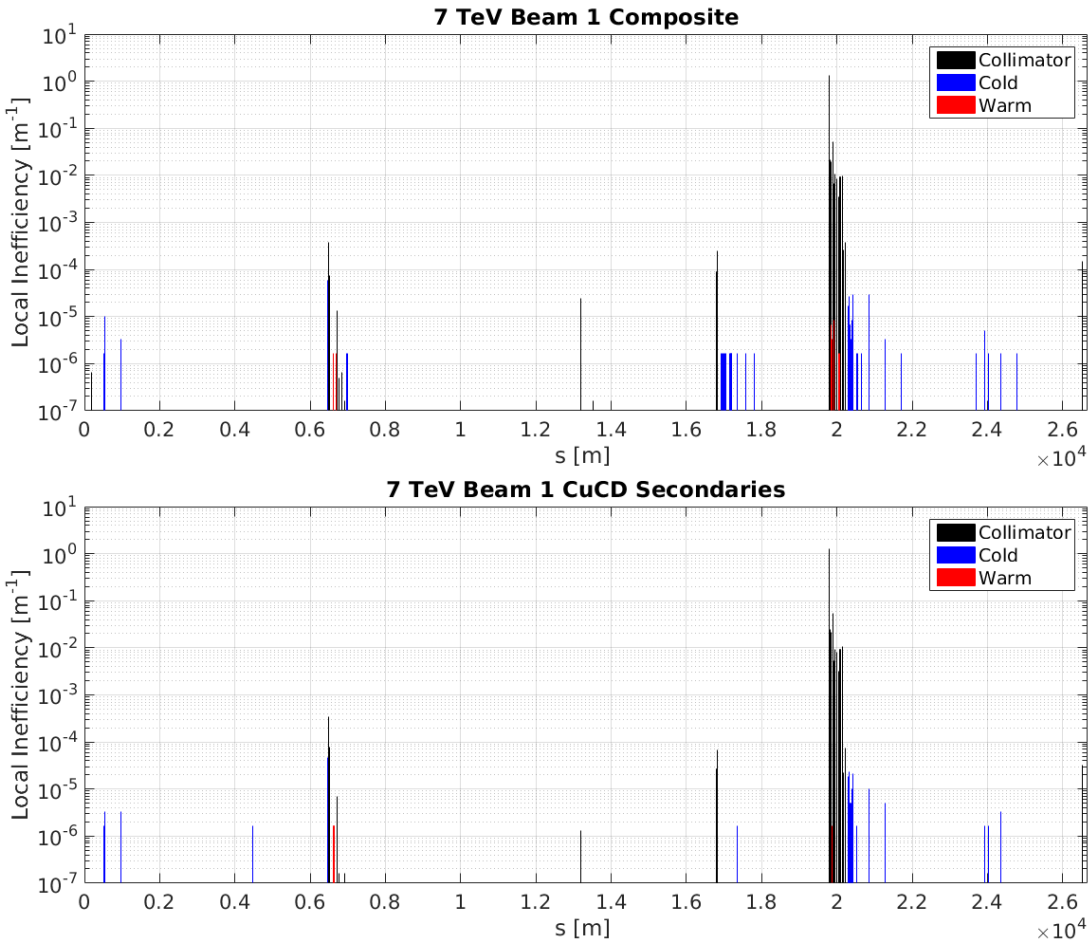

Figure 6.8: Loss map for beam 1 of the nominal LHC, comparing the use of CFC (above) with CuCD (below) for secondary collimator jaws. We observe a decrease in cold losses throughout the machine when using CuCD secondary collimators.

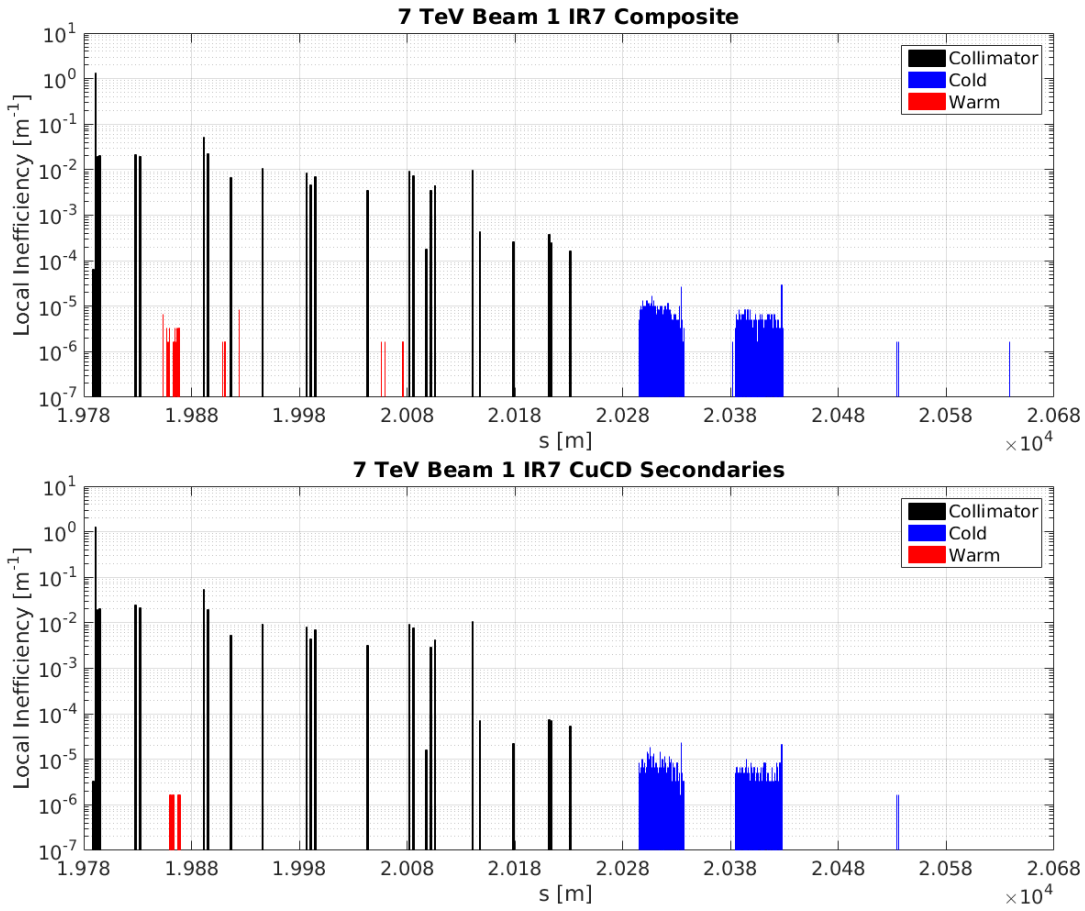

Figure 6.9: Loss map for beam 1 of the nominal LHC in IR7, comparing the use of CFC (above) with CuCD (below) for secondary collimator jaws. We observe a decrease in cold and absorber losses in IR7 due to the increase in inelastic scattering events in CuCD secondary collimators as compared to CFC.

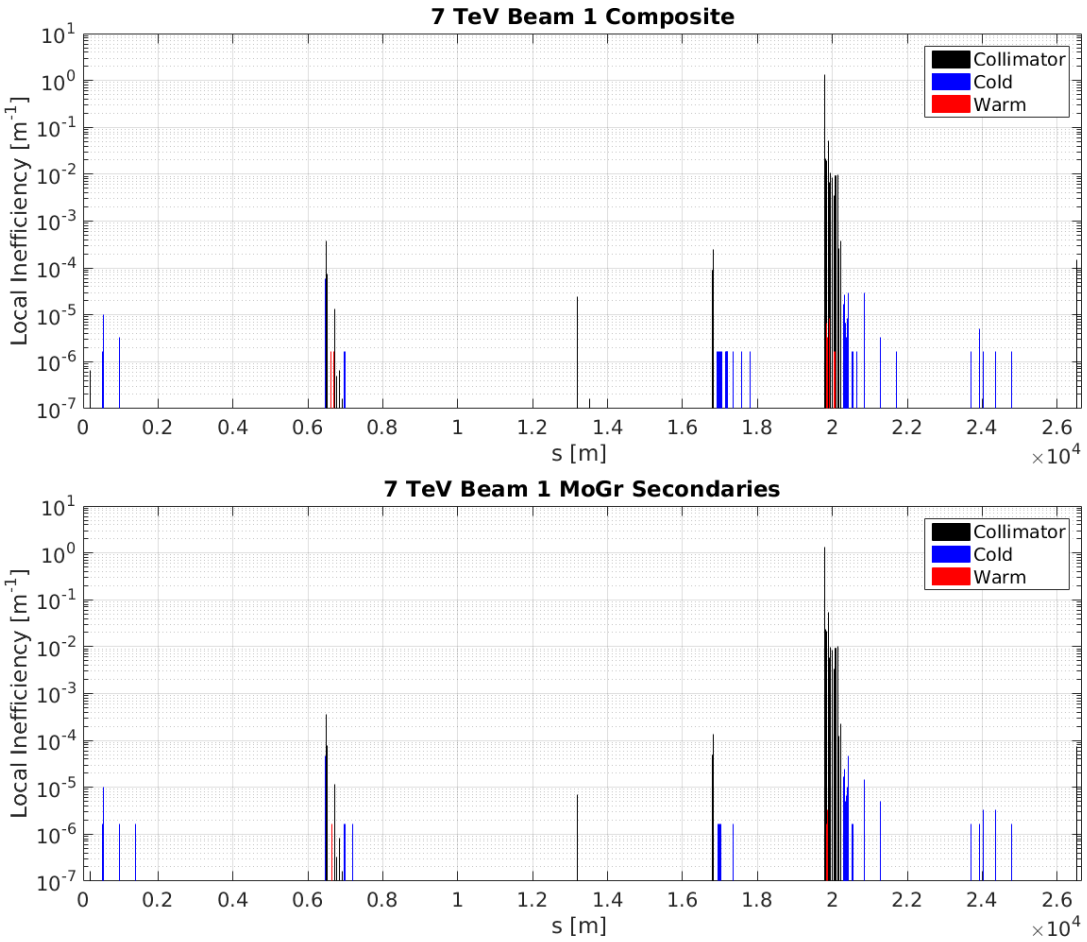

Figure 6.10: Loss map for beam 1 of the nominal LHC, comparing the use of CFC (above) with MoGr (below) for secondary collimator jaws. There appears to be less of a difference in cold losses when compared to using CuCD for secondary collimator jaws.

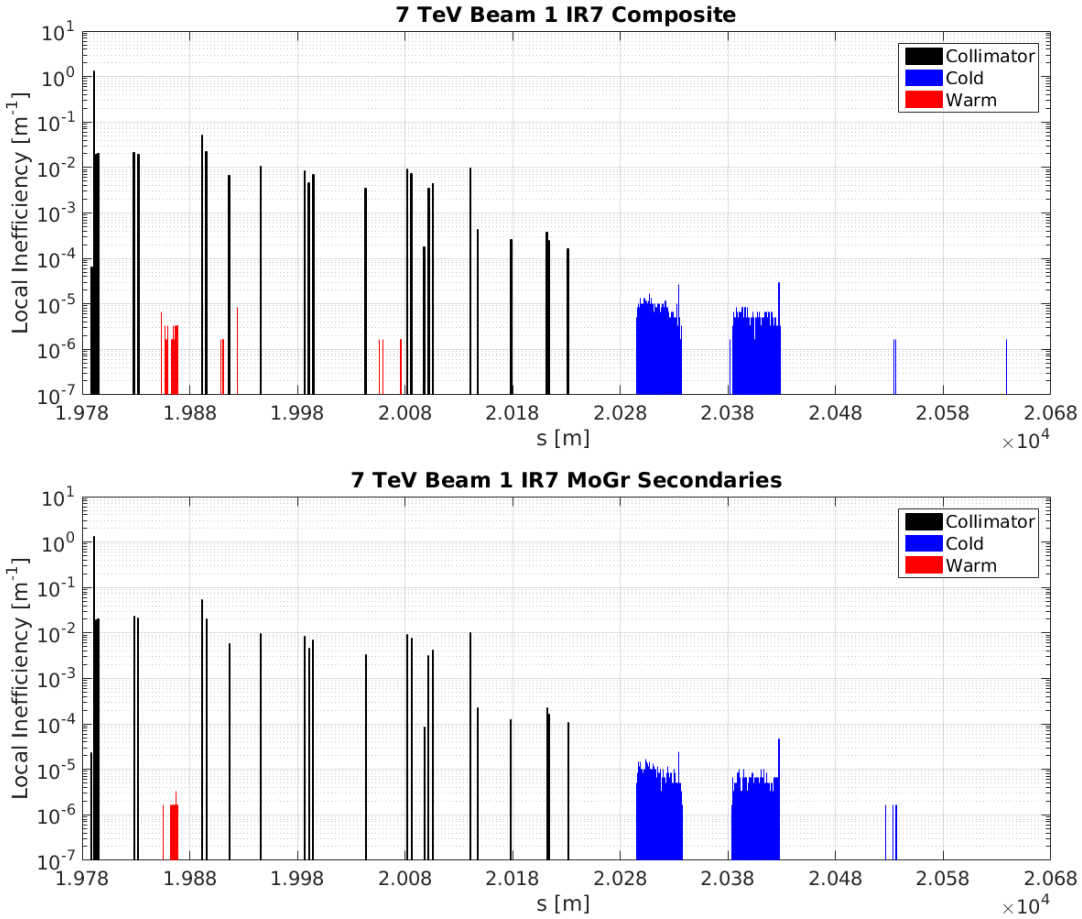

Figure 6.11: Loss map for beam 1 of the nominal LHC in IR7, comparing the use of CFC (above) with MoGr (below) for secondary collimator jaws. Similar to CuCD we observe a reduction in cold and absorber losses due to an increase in inelastic scattering events in MoGr secondary collimators. This is not as pronounces as when using CuCD however.

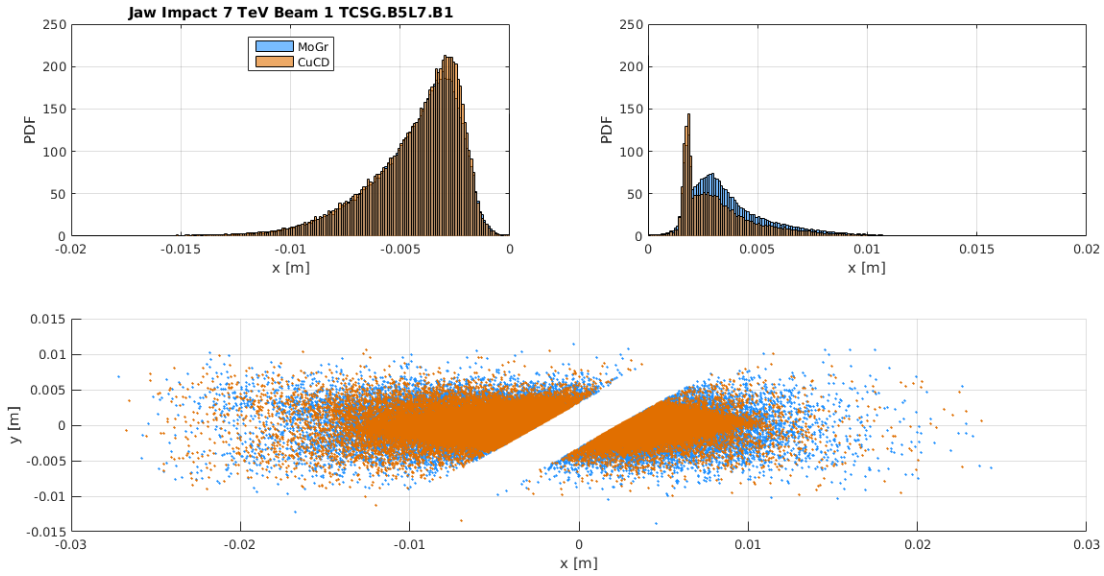

Figure 6.12: Particle impact distribution in the transverse plane for the most loaded secondary collimator TCSG.B5L7.B1, comparing the use of CuCD and MoGr for primary collimator jaws. The inside jaw receives more impacts, and the novel materials give different impact distributions.

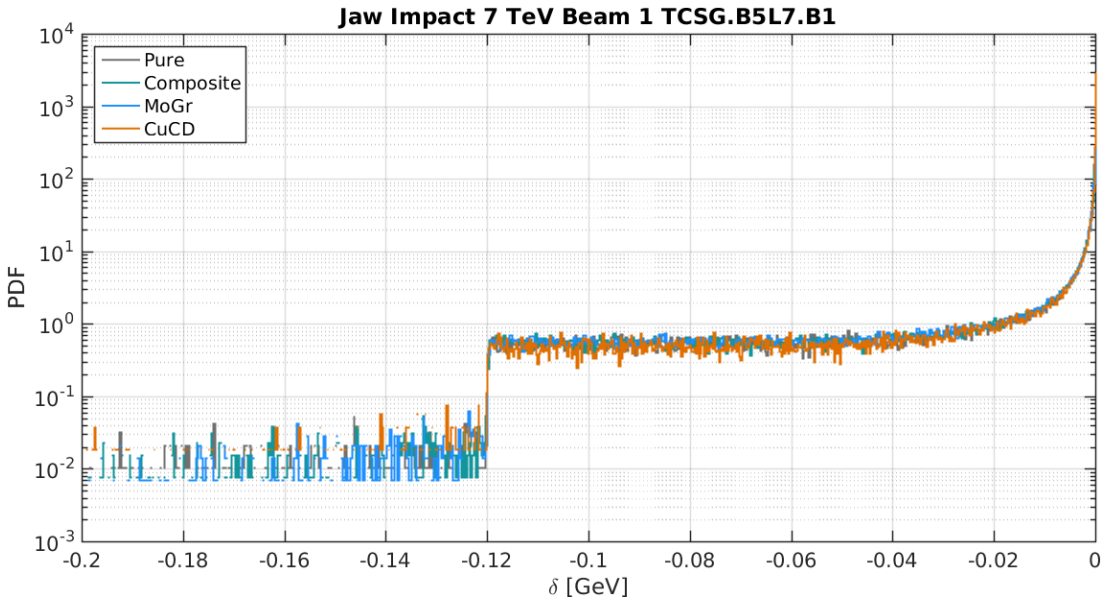

Figure 6.13: Particle impact distribution in dp at the most loaded secondary collimator TCSG.B5L7.B1, comparing the use of pure elements, standard composites, CuCD, and MoGr for primary collimator jaws. All of these distributions are similar, and show the characteristic cut-off from single diffractive scattering in the primary collimator.

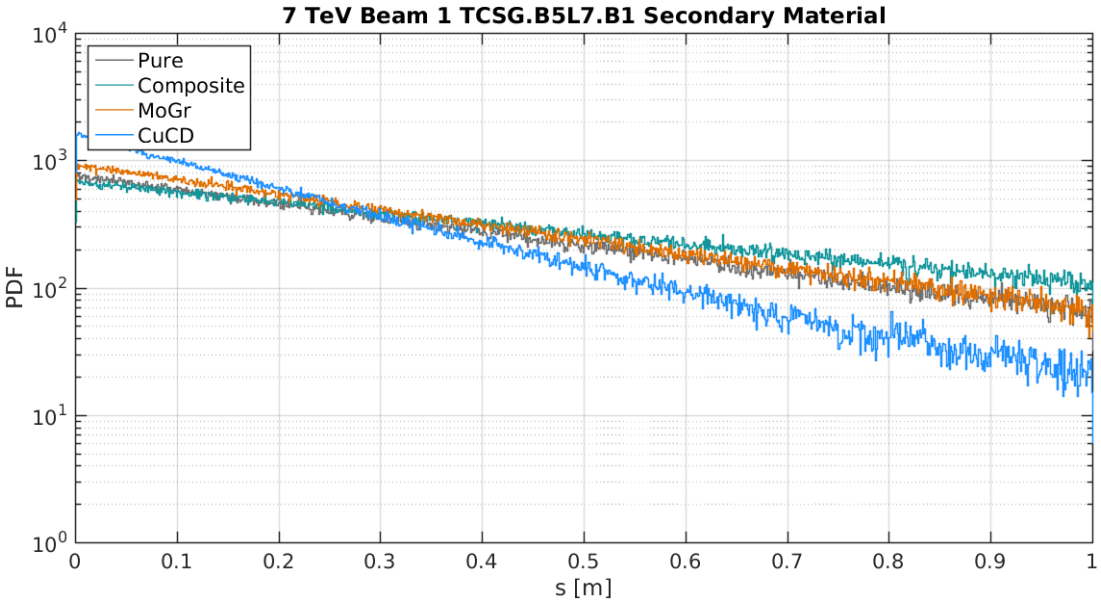

Figure 6.14: Loss distribution in s at the most loaded secondary collimator TCSG.B5L7.B1, comparing the use of pure elements, standard composites, CuCD, and MoGr for secondary collimator jaws. CuCD has the shortest mean free path, giving a loss distribution skewed more towards the start of the collimator.

Region	**Type**	**Losses**			
		MoGr Primary	**CuCD Primary**	**MoGr Secondary**	**CuCD Secondary**
Entire LHC	**Collimator**	6133915	6155375	6077073	6076465
	Cold	1519	836	1936	1749
	Warm	54	58	16	4
IR7	**Collimator**	6127191	6150336	6072520	6073432
	(Primary)	5151768	5458136	4902892	4899948
	(Secondary)	966973	685110	1164456	1171704
	(Absorber)	8450	7090	5172	1780
	Cold	1375	740	1791	1609
	Warm	54	57	15	3
IR3	**Collimator**	1522	850	1916	1819
	Cold	52	28	51	51
	Warm	0	1	1	1

Table 6.3: Breakdown of losses in loss maps for beam 1 of th 7 TeV LHC, comparing the use of novel composites MoGr and CuCD for primary or secondary collimator jaw materials in MERLIN.

		MoGr Primary	CuCD Primary	MoGr Secondary	CuCD Secondary
IR7	**Collimator**	+1%	+1%	+0%	+0%
	(Primary)	+5%	+11%	+0%	+0%
	(Secondary)	-15%	-40%	+1%	+2%
	(Absorber)	-5%	+26%	-42%	-80%
	DS	-22%	-48%	+1%	-9%

Table 6.4: Percentage change in losses for beam 1 of the 7 TeV LHC, comparing the use of novel composites as primary or secondary collimator jaw materials, with the standard composites currently used.

Insertion Region	Collimator Family	Setting [σ]
7	Primary	6
	Secondary	7
	Absorber	10
	TCLD	999 / 10
3	Primary	15
	Secondary	18
	Absorber	20
1	Tertiary	8.3
	Absorber	10
5	Tertiary	8.3
	Absorber	10
6	Dump Protection	8
	Secondary	7
2	Tertiary	8.3
8	Tertiary	8.3

Table 6.5: HL-LHC collimation settings used for loss map simulations.

6.1.2 HL-LHC

We begin by observing the effect of standard and novel composite materials on loss map results, and then move on to look at the implementation of TCLD dispersion suppressor collimators, before studying the hollow electron lens as a collimation enhancer. Version 1.2.1 of the beam 1 HL-LHC lattice was used for these simulations, with collision optics and two thick lens TCLDs in IR7. The horizontal beta and dispersion functions calculated in MERLIN are compared with that of those calculated in MADX in Fig. 6.15 and 6.16 respectively. MERLIN is identical to MADX as expected. The collimator settings used for these loss map simulations are detailed in Table 6.5. The initial distribution used is similar to that in the 7 TeV simulations, a horizontal halo with no longitudinal component and an impact parameter of 1 μm, as shown in Figures 6.17 to 6.19.

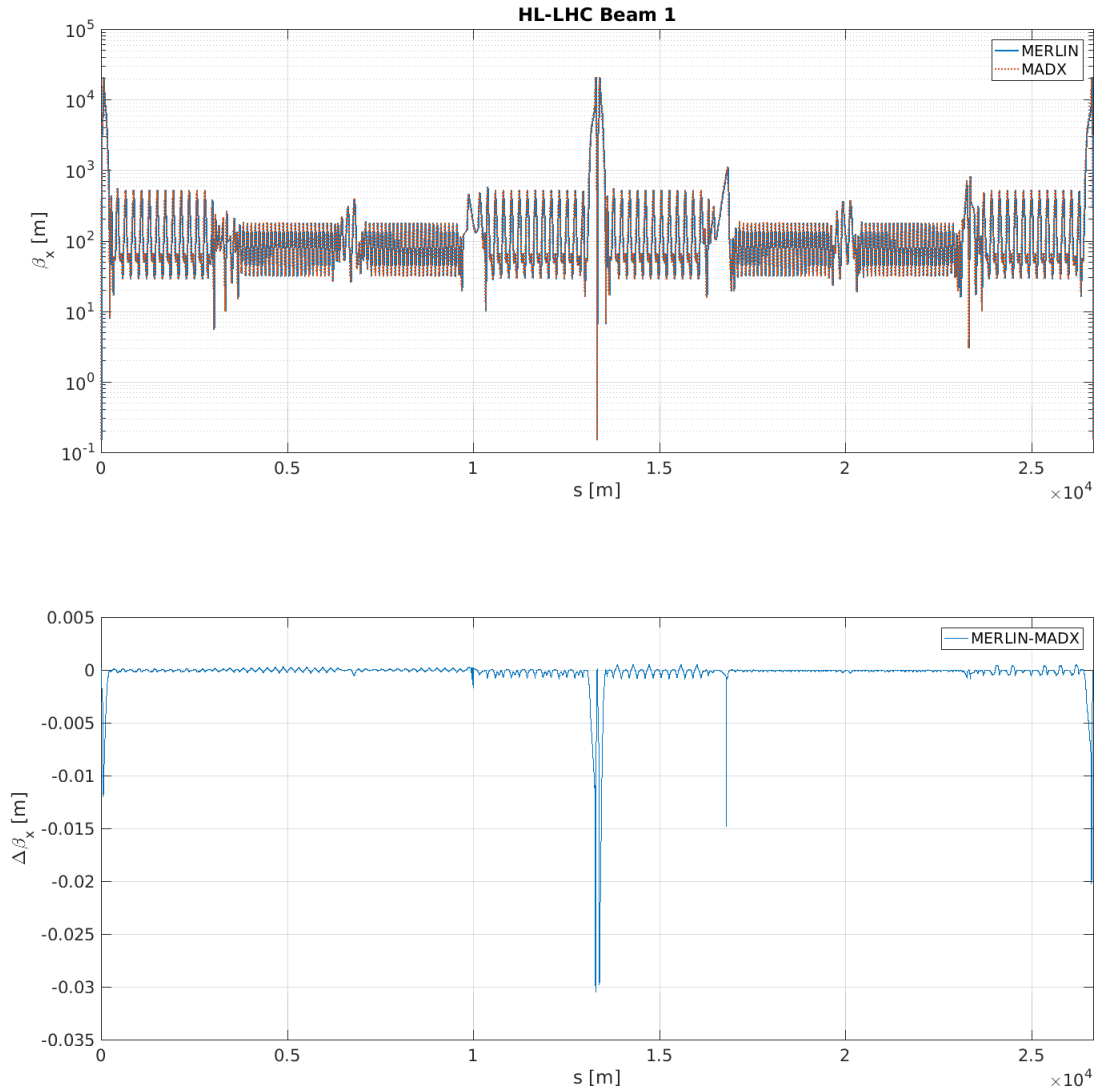

Figure 6.15: β_x function for beam 1 of the HL-LHC, MERLIN (blue) is compared to MADX (orange)in the top plot. The bottom plot shows the difference between MERLIN and MADX, which is made using an interpolation algorithm. The difference is small, and is due to the number of significant figures used in the data.

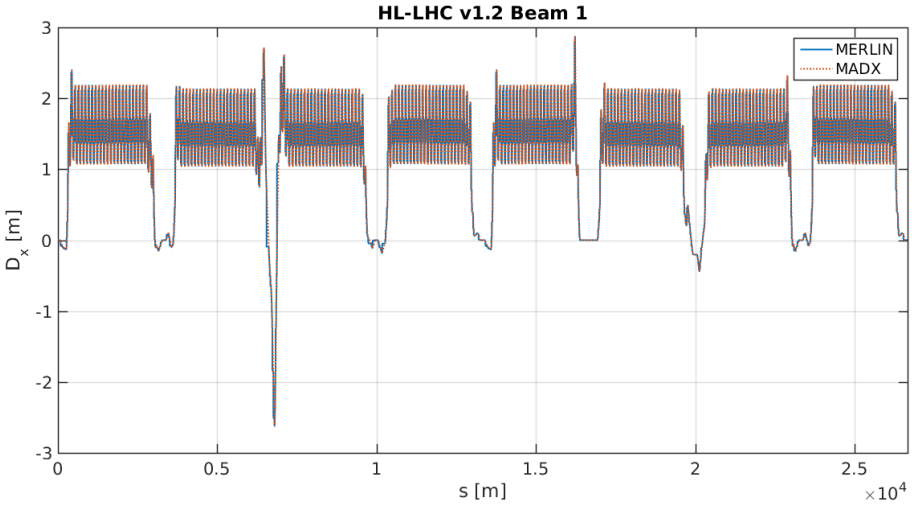

Figure 6.16: Dispersion D_x function for beam 1 of the HL-LHC, MERLIN (blue) is compared to MADX (orange).

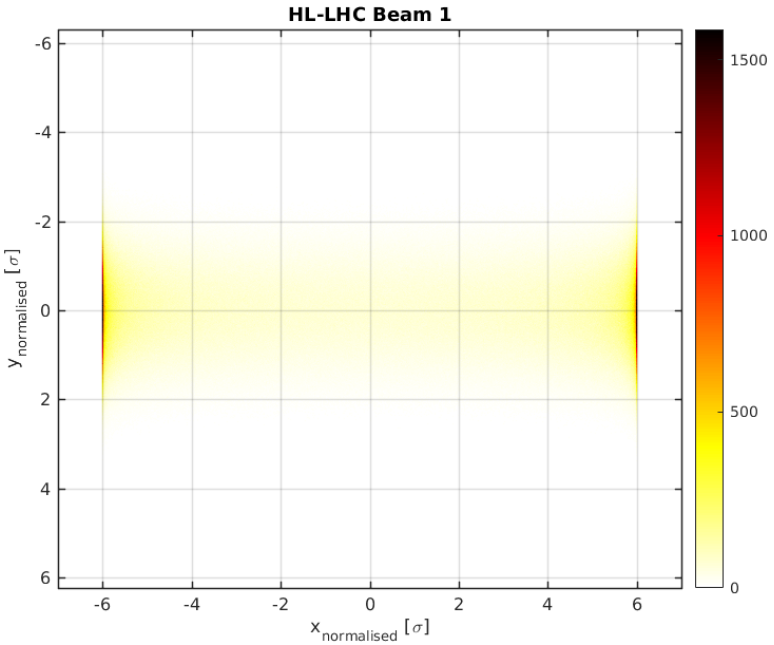

Figure 6.17: Initial distribution for the HL-LHC beam 1 simulations, showing normalised transverse space.

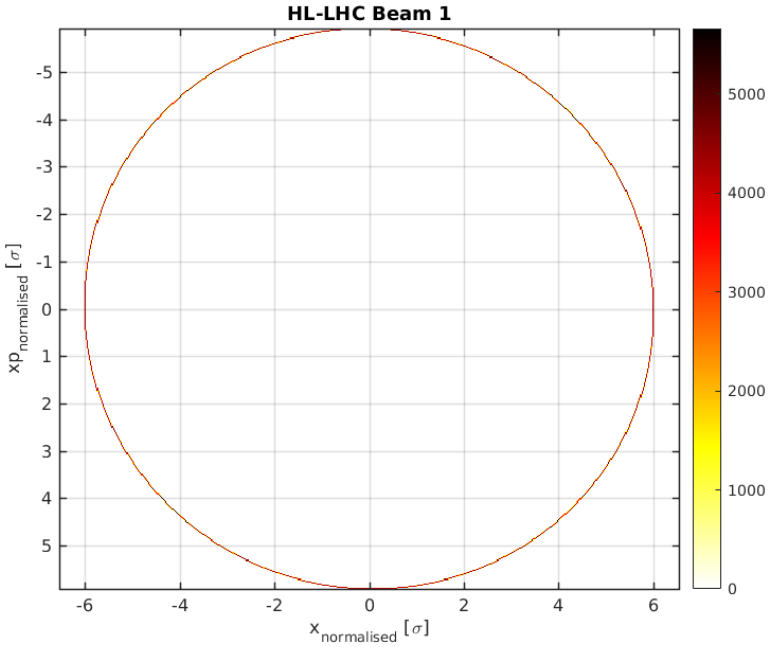

Figure 6.18: Initial distribution for the HL-LHC beam 1 simulations, showing normalised xx' phase space.

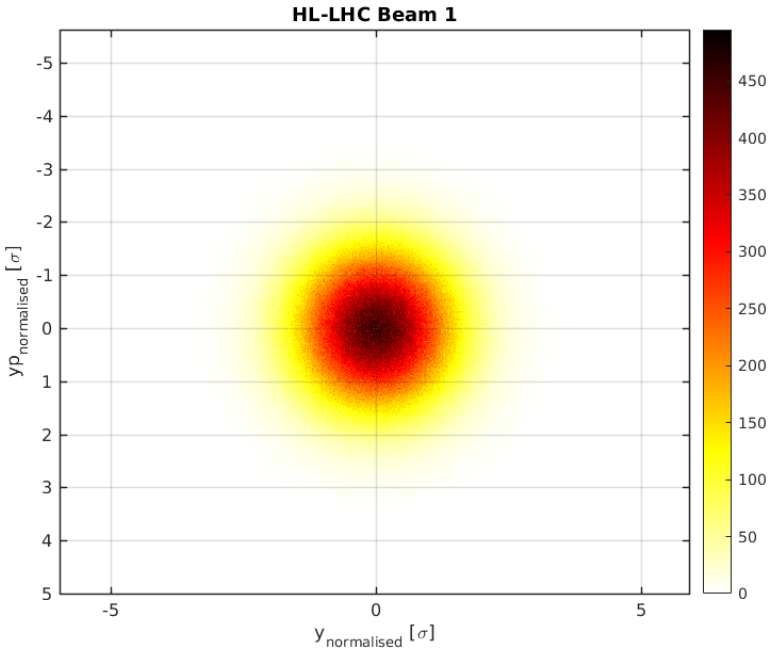

Figure 6.19: Initial distribution for the HL-LHC beam 1 simulations, showing normalised yy' phase space.

6.1.2.1 Effect of Composite Materials

Once more we compare the effect of using composite materials with pure elements for collimator jaws. Figure 6.20 compares the loss maps for the full LHC, and Fig. 6.21 shows IR7. The differences are much the same as in the 7 TeV case: there are more total losses with pure elements, $\approx 30\%$ more cold losses when using composites, as well as more primary losses and twice the absorber losses in IR7. More losses occur in IR3 with composites, as shown in Table 6.6 which gives a full breakdown of losses.

The transverse impact distribution on the most loaded secondary collimator in IR7 is shown in Fig. 6.22, and the *dp* distribution is shown in Fig. 6.23. These show the same trends as in the nominal LHC, with a small shift towards the most loaded jaw when using pure elements, and the opposite when using composites. The *dp* distribution shows how the new single diffractive algorithm provides the hallmark of particles that have undergone such an interaction in the primaries and escape to dominate this collimators impacts.

The differences between the secondary halo escaping the pure and composite primary collimators in IR7 and impacting upon this collimator are very similar to the nominal LHC case. A 16% increase in IR7 secondary collimator losses occurs when switching pure elements with composite materials. This is smaller than the 26% in 6.5 TeV, and 18% in nominal loss maps. The use of composite materials reveals larger cold losses in the dispersion suppressor magnets following the betatron collimation insertion. This is important when considering the dispersion suppressor collimators (TCLDs), which are added in the next section.

		Pure	**Composite**
Entire LHC	**Collimator**	6353351	6330894
	Cold	3060	3915
	Warm	15	39
IR7	**Collimator**	6298590	6269523
	(Primary)	5250714	5046364
	(Secondary)	1042435	1213363
	(Absorber)	5441	9796
	Cold	2174	2585
	Warm	14	37
IR3	**Collimator**	968	1195
	Cold	0	4
	Warm	1	2

Table 6.6: Breakdown of losses in loss maps for beam 1 of the HL-LHC, comparing the use of pure elements and composites for collimator jaw materials.

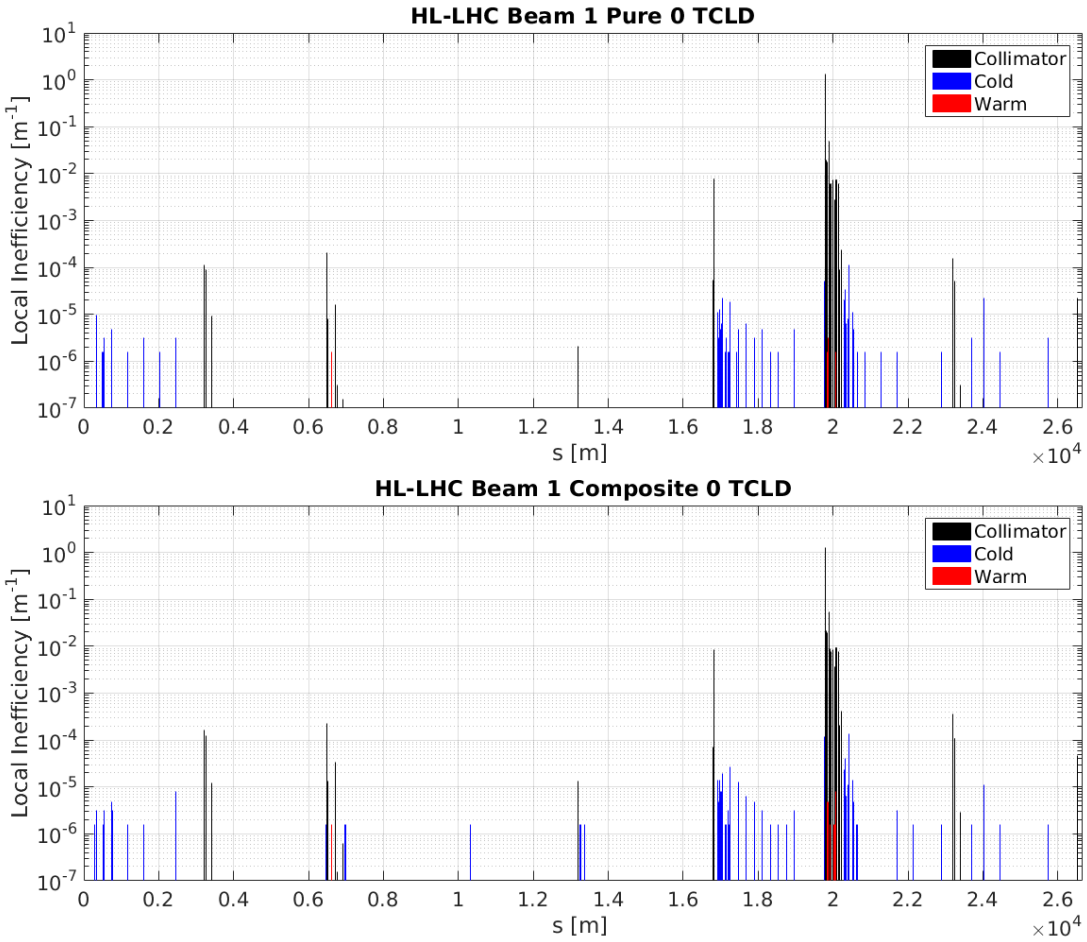

Figure 6.20: HL-LHC loss maps comparing the use of pure elements (above) with composite materials (below) for collimator jaws. The losses appear quite similar, though there are differences which are summarised in Table 6.6.

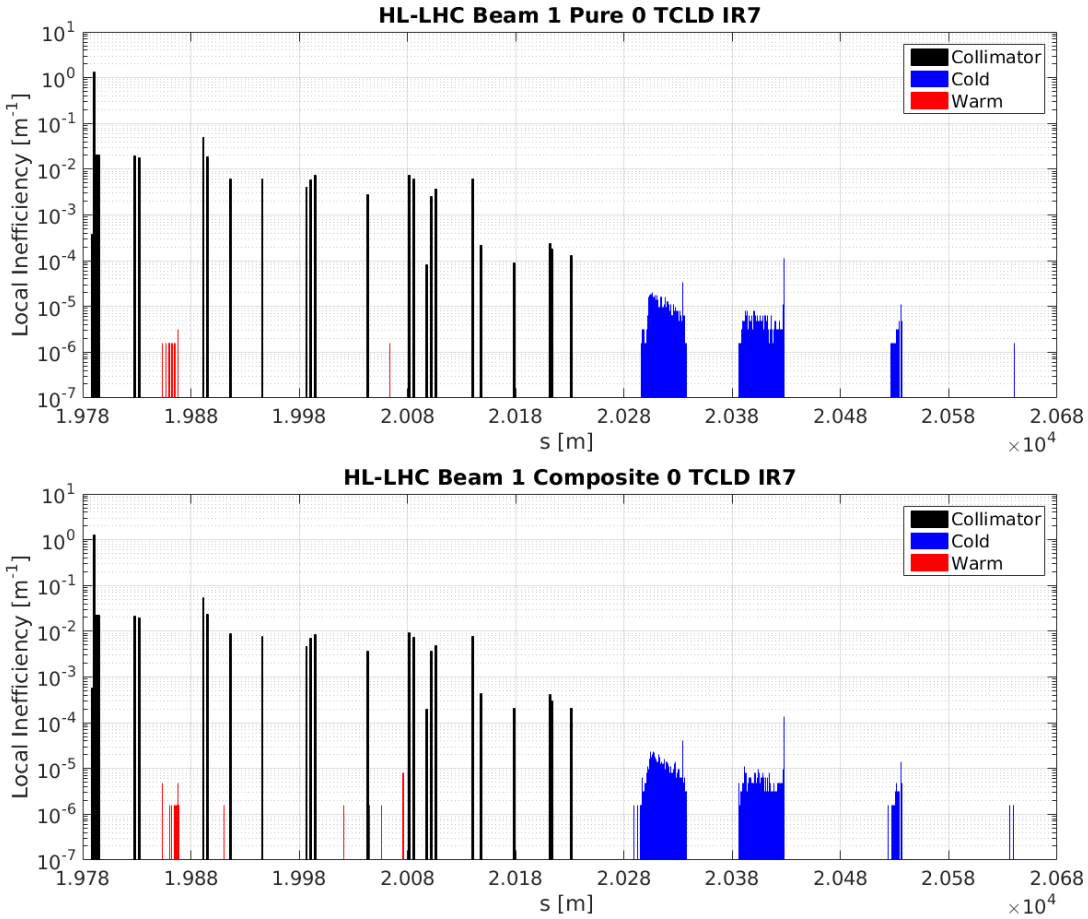

Figure 6.21: HL-LHC loss maps comparing the use of pure elements (above) with composite materials (below) for collimator jaws in IR7. When using composites more protons escape the primary collimators, giving rise to an increased loss in secondary collimators, absorbers, and cold elements in the dispersion suppressor region.

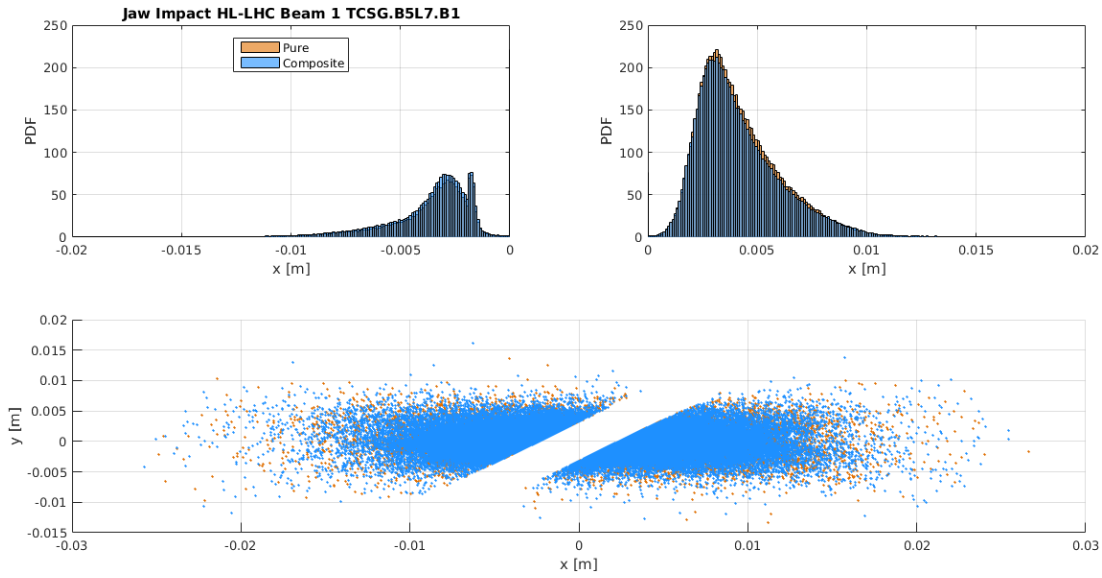

Figure 6.22: Particle impact distribution at the most loaded secondary collimator TCSG.B5L7.B1, comparing the use of pure elements and standard composites for secondary collimator jaws.

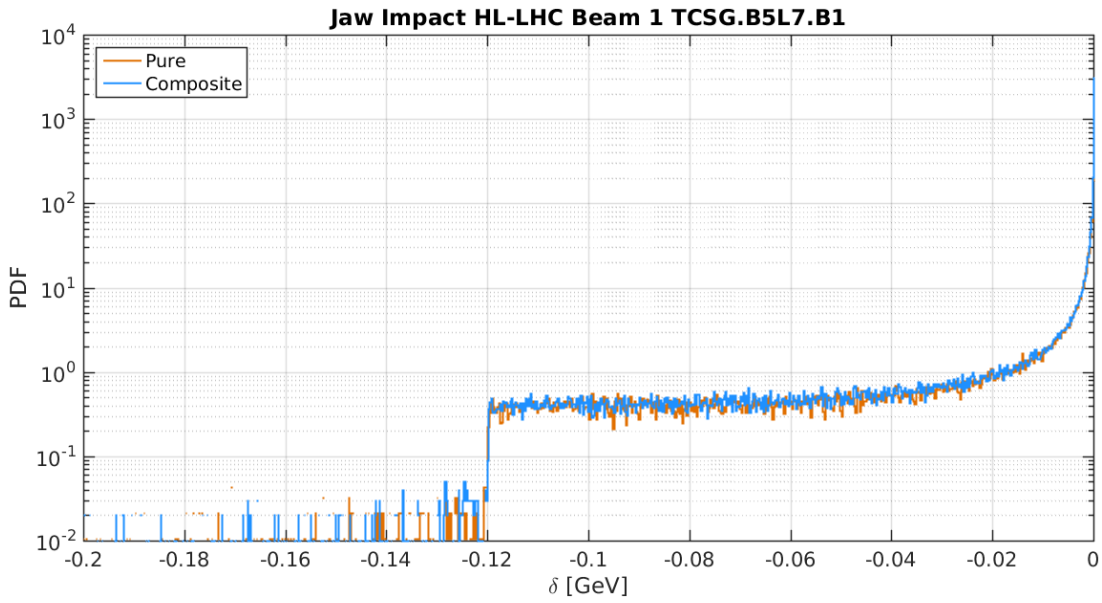

Figure 6.23: Particle impact distribution in dp at the most loaded secondary collimator TCSG.B5L7.B1, comparing the use of pure elements and standard composites for secondary collimator jaws. Again the cut-off is ascribed to single diffractive interactions in the primary collimators.

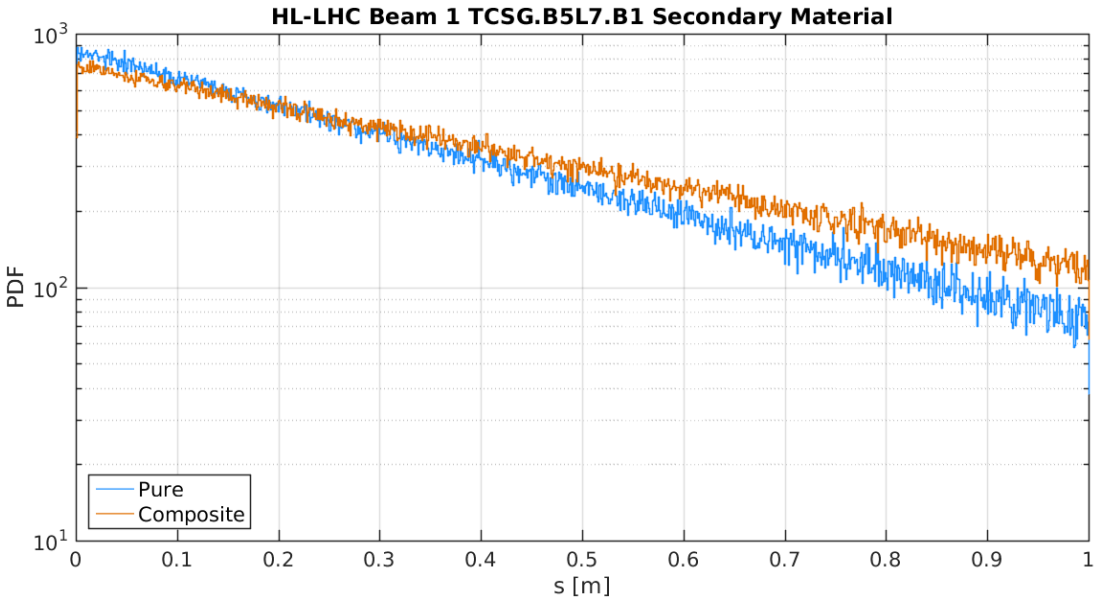

Figure 6.24: Loss distribution in s at the most loaded secondary collimator TCSG.B5L7.B1, comparing the use of pure elements, and standard composites for secondary collimator jaws.

6.1.2.2 Effect of Dispersion Suppressor Collimators

The TCLDs post betatron collimation insertion in IR7 were closed to 10σ in order to predict their effectiveness in reducing cold losses in the DS region. The TCLD jaws are made of Inermet180®. The beta and dispersion functions in x are shown in Fig. 6.25 and Fig. 6.26 respectively, and Table 6.7 details the half gaps in m. The dispersion increases after the collimation insertion (19.8 - 20.2 km), the TCLDs are positioned on this steep rise in order to intercept particles that have undergone single diffractive scattering in the collimation insertion and escape it, as observed in LHC operation.

Using standard composite materials for collimator jaws, the loss maps using 0, 1, and 2 TCLDs are shown in Fig. 6.27, where the use of one TCLD means the first only (TCLD.8R7.B1). It is evident that the presence of one TCLD reduces the cold DS losses and the majority of cold losses in the machine, and the addition of the second TCLD reduces almost all cold losses. In future work the secondary collimator in IR6 should be given a more relaxed jaw opening in order to reduce the impact on this loss map. We also see that momentum collimation losses are reduced greatly with two TCLDs closed, this hints at a violation of the momentum collimation hierarchy as a TCLD becomes the primary momentum collimator. As this loss map is for betatron collimation and not momentum collimation this is not conclusive.

IR7 is shown in Fig. 6.28, in which the reduction in DS losses is clear, there is a 94 % reduction with a single TCLD, and 96 % with both. A breakdown of losses is given in Table 6.8, from which we see that the second TCLD appears to violate the momentum collimation hierarchy.

We observe that the use of two TCLDs post IR7 closed at 10 σ, removes nearly all DS cold losses, and in fact the momentum halo escaping the betatron collimation insertion. This raises a concern however as it appears that the second TCLD acts as the primary momentum collimator, this (under normal operation with an off-momentum halo) is likely to cause more DS losses as particle showers escape the TCLD under high load.

Off-momentum halo studies with TCLDs have been performed [99], the preliminary findings of which indicate that a jaw opening of 12.5 σ does not violate the momentum cleaning hierarchy, however at 10 σ the TCLDs become primary momentum collimators. This has been shown to reduce the efficiency of the momentum collimation system, and cause cold DS losses as a secondary momentum halo is created in a region with no downstream secondary collimators. This is a cause for concern as it limits the effectiveness of the TCLDs, further study is required in this area.

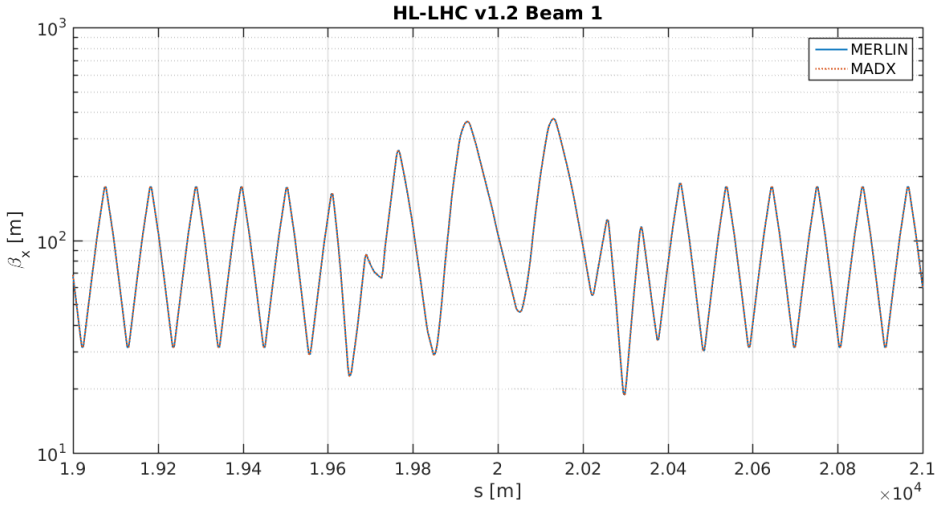

Figure 6.25: β_x function at IR7 for beam 1 of the HL-LHC, MERLIN (blue) is compared to MADX (orange).

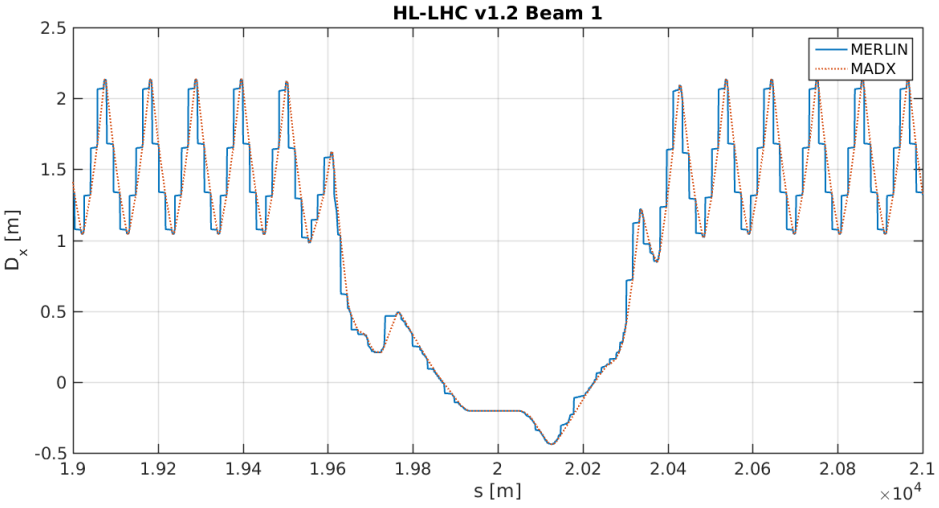

Figure 6.26: Dispersion D_x function at IR7 for beam 1 of the HL-LHC, MERLIN (blue) is compared to MADX (orange).

Parameter	**TCLD.8R7**	**TCLD.10R7**
s $[m]$	20287.06	20366.59
β_x $[m]$	32.34	46.55
β_y $[m]$	126.04	133.2
Half Gap $[\sigma]$	10	10
Half Gap $[m]$	0.00121	0.00146

Table 6.7: HL-LHC collimation settings used for HEL simulations.

Region	**Type**	**Losses**		
		Composite 0 TCLDs	**Composite 1 TCLD**	**Composite 2 TCLDs**
Entire LHC	**Collimator**	6330894	6333244	6333252
	Cold	3915	1272	1231
	Warm	39	58	49
IR7	**Collimator**	6269523	6273292	6275564
	(Primary)	5046364	5047076	5047224
	(Secondary)	1213363	1212470	1211859
	(Absorber)	9796	9771	9639
	(TCLD8)	-	3975	4007
	(TCLD10)	-	-	2835
	Cold	3915	224	160
	Warm	39	53	49
IR3	**Collimator**	1195	1220	11
	Cold	4	3	0
	Warm	2	5	0

Table 6.8: Breakdown of losses in loss maps for beam 1 of the HL-LHC, comparing losses with 0, 1, or 2 TCLD jaws closed to 10 σ, using standard composite materials for collimator jaws. 1 TCLD refers to the first downstream of IR7 - TCLD.8R7.B1, 2 TCLDs refers to both this and the next downstream TCLD.10R7.B1.

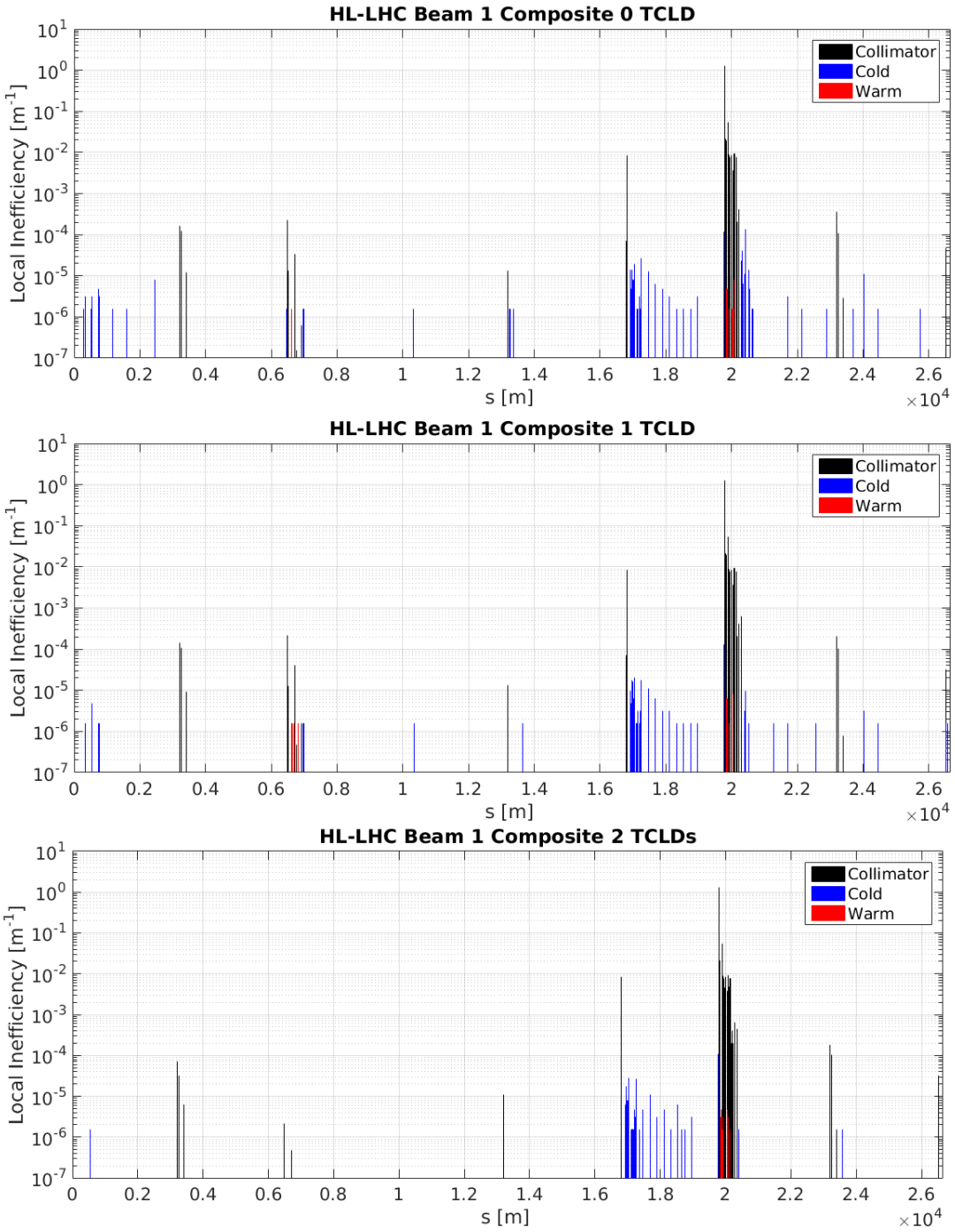

Figure 6.27: HL-LHC loss maps comparing the use of 0, 1, and 2 TCLDs with composite materials for collimator jaws. There is a clear reduction in cold losses from protons escaping the betatron collimation insertion. The reduction of losses in the momentum cleaning insertion (around s = 6500 m) indicates that the momentum cleaning hierarchy has been violated.

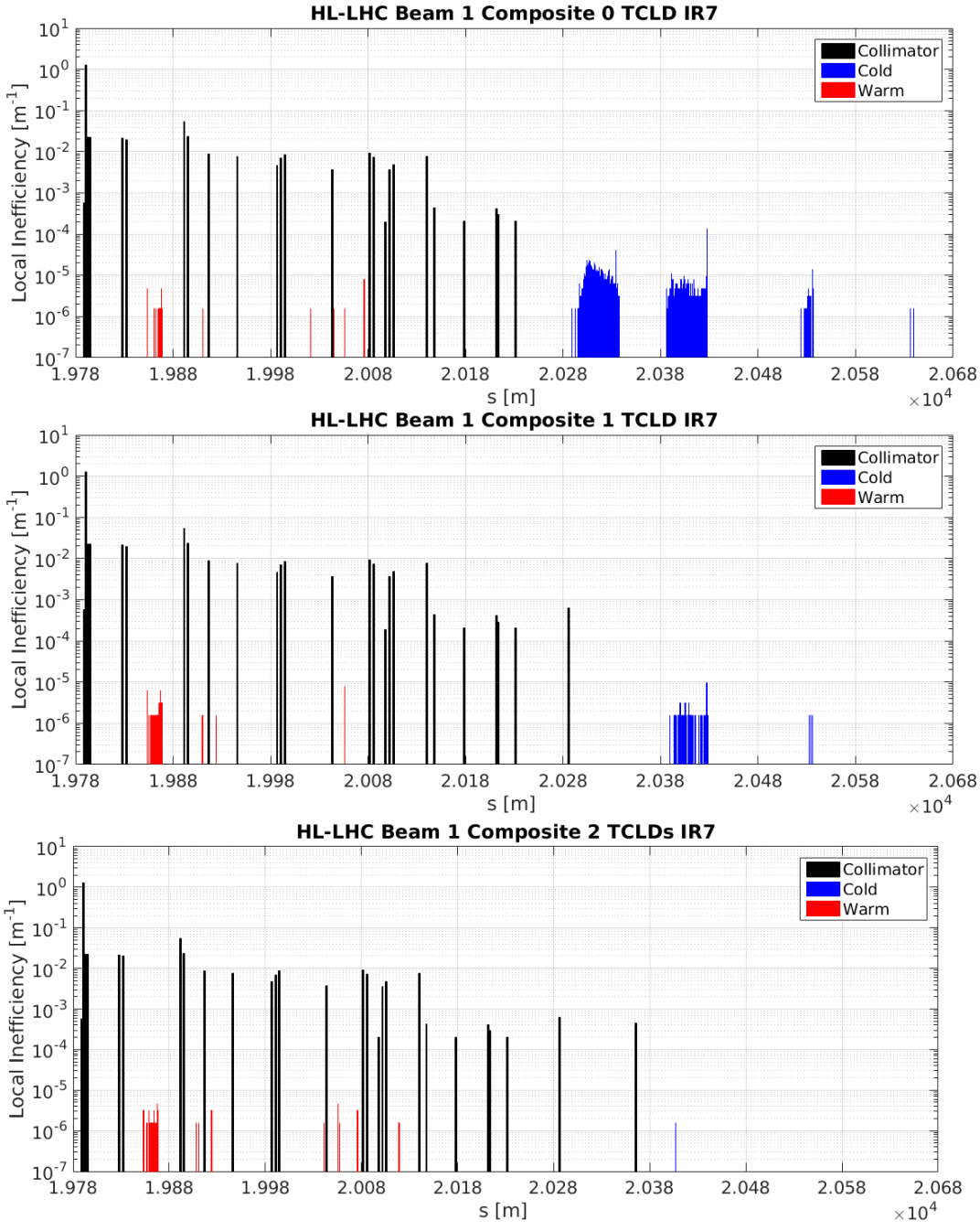

Figure 6.28: HL-LHC loss maps comparing the use of 0, 1, and 2 TCLDs with composite materials for collimator jaws in IR7. We confirm that nearly all cold losses in the dispersion suppressor are mitigated with the use of two TCLDs.

6.1.2.3 Effect of Novel Materials

Novel composites are being investigated in order to provide an efficient HL-LHC collimation insertion. It is envisioned that such materials will replace the current CFC AC150K® secondary collimator jaws. As such they were replaced in MERLIN in order to compare the effect on loss maps, firstly with TCLD jaws open. Full loss maps using the standard composite CFC AC150K®, and novel materials MoGr and CuCD are shown in Fig. 6.29.

These loss maps appear similar, with CuCD resulting in reduced collimator losses at the interaction points 8, 1, and 5. The breakdown of losses is given in Table 6.9, from which it is clear that there is a similar total loss in all cases, which is what we expect from the previous nominal LHC investigation. We see the reduction in IR7 absorber losses when using novel composites, indicating a reduction in the halo escaping the primary and secondary collimators. In the nominal LHC case the use of novel secondary collimator materials did not have a great impact on the cold DS losses in IR7, however in the HL-LHC case there is a clear reduction without the use of TCLDs, with a reduction of 44% for CuCD, and 42% for MoGr. There is no noticeable effect on the momentum collimation insertion losses.

The requirement for novel collimator jaw materials in the HL-LHC is driven by impedance and robustness, for which these materials are believed to offer an improvement over CFC [47]. Here we have shown that MoGr and CuCD offer some improvement over CFC. When using novel composites as secondary collimator jaw materials it is clear that MoGr and CuCD behave similarly, and both materials offer comparable efficiency to the CFC AC150K® jaws currently used, with a significant reduction in the cold DS losses post collimation. The cold DS magnets are the loss location of greatest concern in the LHC, and with higher luminosity their protection is a necessity. Therefore we conclude that MoGr and CuCD are suitable for secondary jaw materials in the HL-LHC, assuming they offer improved mechanical properties (*i.e.* related to robustness and impedance) over CFC.

Region	Type	Losses		
		Composite 0 TCLDs	**CuCD 0 TCLDs**	**MoGr 0 TCLDs**
Entire LHC	**Collimator**	6330894	6324299	6322987
	Cold	3915	2995	3140
	Warm	39	3	18
IR7	**Collimator**	6269523	6289275	6284774
	(Primary)	5046364	5047412	5050741
	(Secondary)	1213363	1239002	1227394
	(Absorber)	9796	2861	6639
	Cold	3915	2206	2281
	Warm	39	0	16
IR3	**Collimator**	1195	1033	1067
	Cold	4	3	1
	Warm	2	3	2

Table 6.9: Breakdown of losses in loss maps for beam 1 of the HL-LHC, comparing losses standard composite materials, and the novel composites CuCD and MoGr for collimator jaws. TCLDs are open in these simulations.

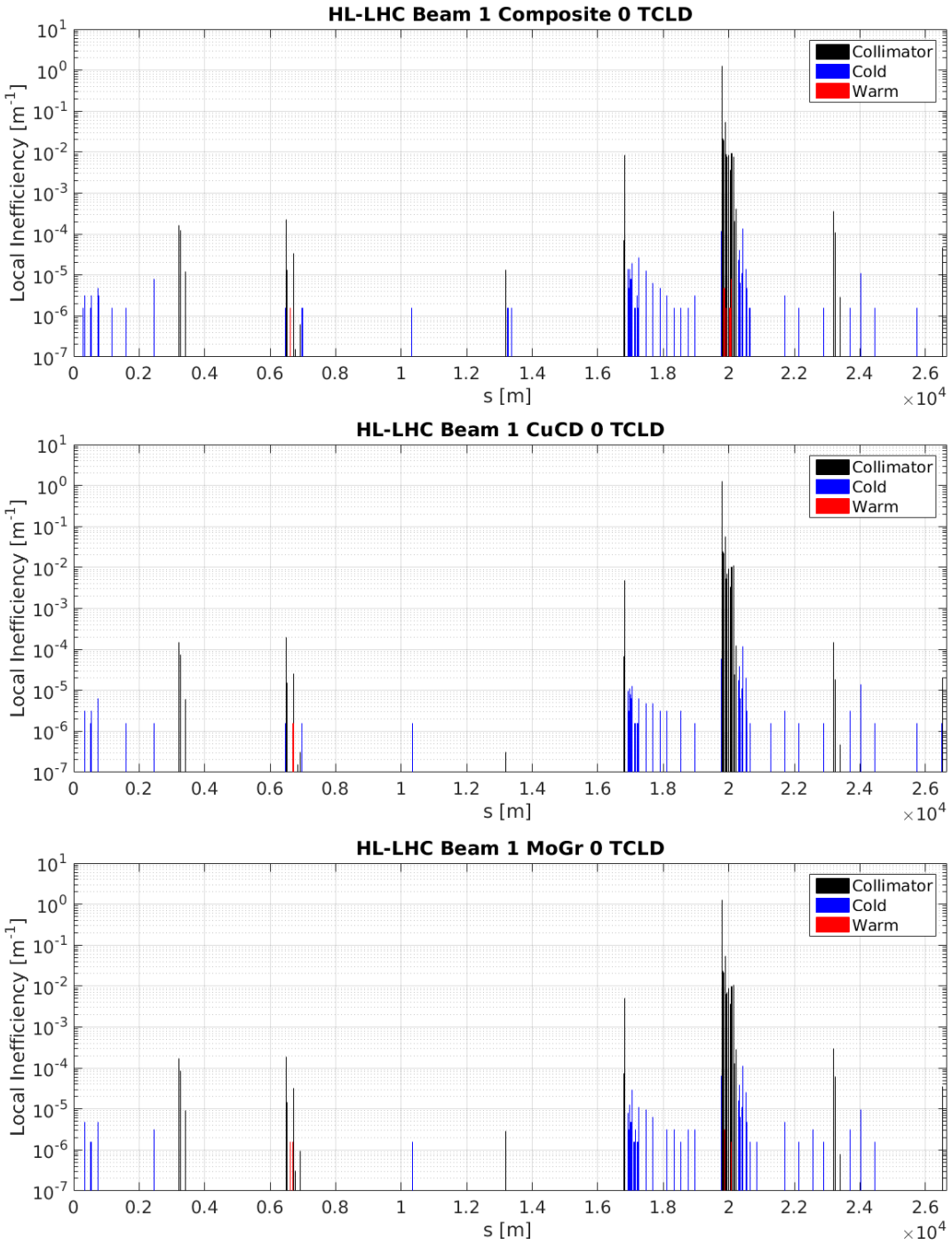

Figure 6.29: HL-LHC loss maps comparing AC150K, CuCD, and MoGr composite materials for collimator jaws, with TCLDs open. As shown in Table 6.9 MoGr and CuCD reduce the halo leakage from IR7 by a small amount when compared to CFC, resulting in fewer cold losses throughout the machine.

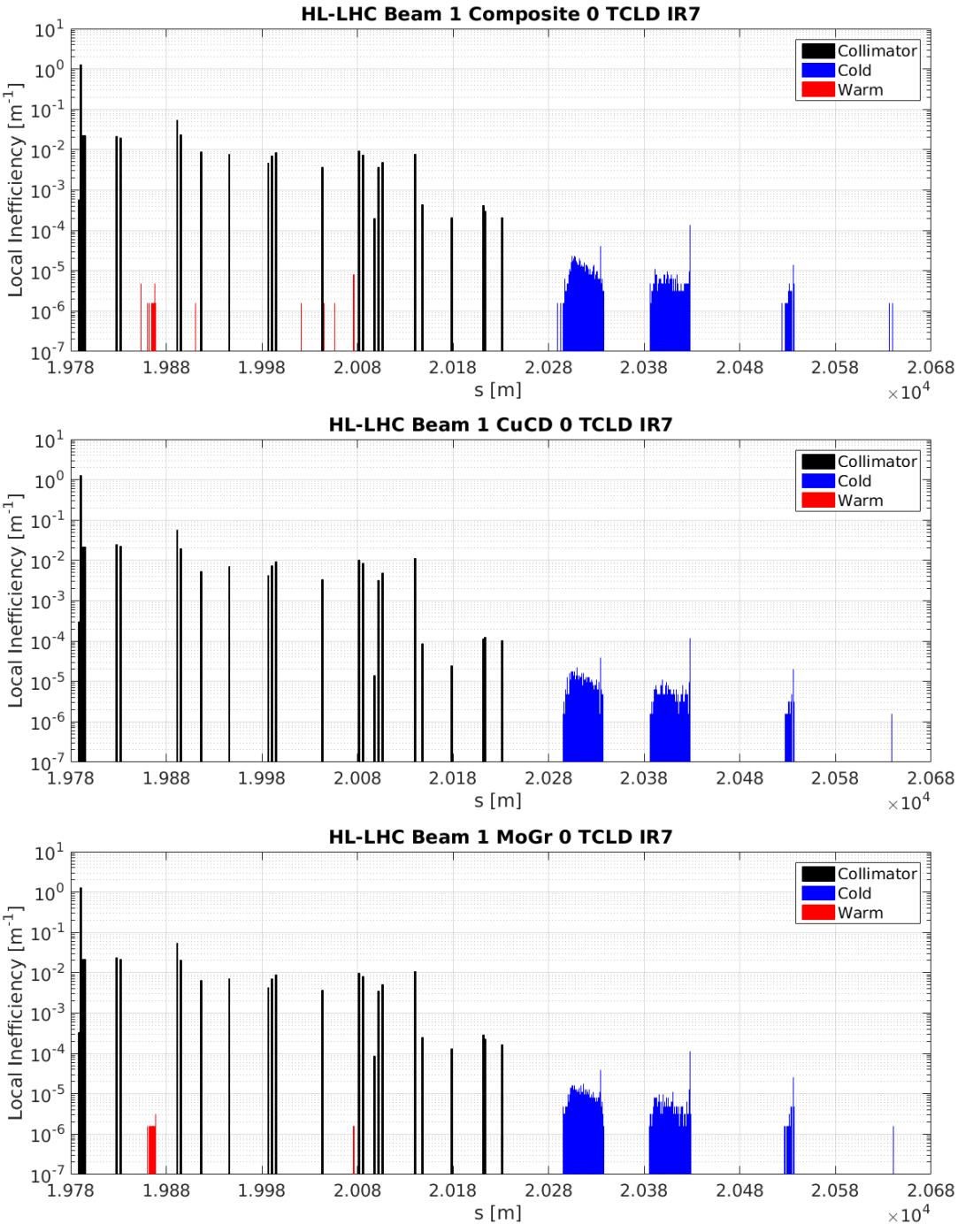

Figure 6.30: HL-LHC loss maps comparing AC150K, CuCD, and MoGr composite materials for collimator jaws, with TCLDs open, showing IR7. Though it is not evident, the total cold loss in IR7 is reduced by *approx* 40% when using the novel composites MoGr or CuCD for secondary collimators when compared to CFC.

6.1.2.4 Effect of TCLDs with Novel Materials

Finally we add the TCLD collimators in IR7 whilst using novel materials for secondary collimators. First we consider CuCD, the full loss maps with 0, 1, and 2 TCLDs are shown in Fig. 6.31, and IR7 in Fig. 6.32. When comparing to the loss maps with standard composites in Fig. 6.20 we note a clear reduction in the IP5 tertiary collimators, the IR6 secondary collimator, and IP8 tertiary. These are all due to the reduction in halo particles escaping the betatron collimation insertion. In IR7 the loss pattern in the collimation insertion is identical until the DS region as expected. Compared to the standard composite materials fewer particles are lost in the TCLDs, indicating a smaller load on these collimators. As in the standard composite case, one TCLD reduces the cold DS losses greatly, though fewer particles escape the betatron collimation insertion. The reduction is 90% and 97% for one and two TCLDs, which is similar to the 94% and 96% in the standard composite case.

In the case of MoGR secondary collimators, the full loss maps are shown in Fig. 6.33, and IR7 in Fig. 6.34. There are again reductions in collimator losses in IR8, IR1, and IR6 when compared to the standard composite loss maps, however not as significant as the CuCD loss maps. As more particles escape the betatron collimation insertion as compared to CuCD, there are more TCLD losses. The reduction in cold DS losses in IR7 is 90% and 96% which is almost identical to CuCD.

The impact distributions for the TCLDs in cell 8 and 10 of IR7 are shown in Fig. 6.35 and Fig. 6.36 respectively. Comparing the three materials (where composite is CFC AC150K) there appears to be little difference on the negative jaw, with fewest impacts on the TCLD.8R7 positive jaw when using MoGr. It is interesting to note that in both cases the majority of particle impacts occur on the negative jaw, for beam 1 positive x points outwards, thus the single diffractive losses will be on the negative jaw due to reduced rigidity. In the case of the cell 10 TCLD it appears that a one sided collimator may be sufficient for proton cleaning in this case, which could reduce contributions to the machine impedance. The transverse impacts have a range of a few mm, which is much larger than the previously examined secondary collimator which had a range of a few cm. This means that the alignment and jaw gap settings for these collimators are extremely important. Together with the concern of momentum cleaning hierarchy violation these collimators require special attention in future studies.

The loss distributions in s and dp are shown in Fig. 6.37 and Fig. 6.38 respectively. Both show that the losses are more concentrated at the start of the collimator when compared to the previously investigated most loaded secondary IR7 collimator. Though

these simulations use a 1 m active length it is likely that due to space constrains between two 11 T magnets the actual TCLD active length will be 0.6 m, which should be sufficient according to these plots. There appears to be little influence from secondary collimator material, in reality there is a difference in the number of losses, the most with CFC AC150K, then MoGr, and finally CuCD.

As in the standard composite case, the violation of the momentum collimation hierarchy is hinted at with novel materials. A breakdown of CuCD losses is shown in Table 6.10, and MoGr in Table 6.11. In all cases the total number of losses increase as TCLDs are added, and the reduction in cold losses is similar. This indicates that MoGr and CuCD are viable alternatives to secondary collimator jaw materials, though as previously mentioned, other material aspects are under study. It appears that CuCD offers a slight benefit over MoGr as more particles are cleaned in IR7.

In conclusion the novel composite secondary collimators show a reduction in the halo escaping from the betatron collimation insertion when compared to existing CFC jaws, and CuCD provides the greatest improvement in this sense. The use of 2 TCLDs is preferable in terms of reducing cold losses in the machine which are due to the escaping halo from the betatron collimation insertion, however there is a concern for the momentum collimation hierarchy at a TCLD jaw opening of 10 σ.

Region	**Type**	**Losses**		
		CuCD 0 TCLDs	**CuCD 1 TCLD**	**CuCD 2 TCLDs**
Entire LHC	**Collimator**	6324299	6325878	6326852
	Cold	2995	869	678
	Warm	3	6	1
IR7	**Collimator**	6289275	6291183	6294364
	(Primary)	5047412	5047001	5047996
	(Secondary)	1239002	1238678	1238312
	(Absorber)	2861	2844	2818
	(TCLD8)	-	2660	2640
	(TCLD10)	-	-	2598
	Cold	2206	227	67
	Warm	0	4	1
IR3	**Collimator**	1033	1009	14
	Cold	3	0	0
	Warm	3	2	0

Table 6.10: Breakdown of losses in loss maps for beam 1 of the HL-LHC, comparing losses with 0, 1, or 2 TCLD jaws closed to 10 σ, using CuCD for secondary collimator jaws. 1 TCLD refers to the first downstream of IR7 - TCLD.8R7.B1, 2 TCLDs refers to both this and the next downstream TCLD.10R7.B1.

Region	Type	Losses		
		MoGr 0 TCLDs	MoGr 1 TCLD	MoGr 2 TCLDs
Entire LHC	**Collimator**	6322987	6325458	6325765
	Cold	3140	937	732
	Warm	18	14	15
IR7	**Collimator**	6284774	6287967	6290726
	(Primary)	5050741	5051608	5050581
	(Secondary)	1227394	1226329	1227575
	(Absorber)	6639	6686	6614
	(TCLD8)	-	3344	3240
	(TCLD10)	-	-	2716
	Cold	2281	224	101
	Warm	16	13	15
IR3	**Collimator**	1067	1049	15
	Cold	1	1	0
	Warm	2	1	0

Table 6.11: Breakdown of losses in loss maps for beam 1 of the HL-LHC, comparing losses with 0, 1, or 2 TCLD jaws closed to 10 σ, using MoGr for secondary collimator jaws. 1 TCLD refers to the first downstream of IR7 - TCLD.8R7.B1, 2 TCLDs refers to both this and the next downstream TCLD.10R7.B1.

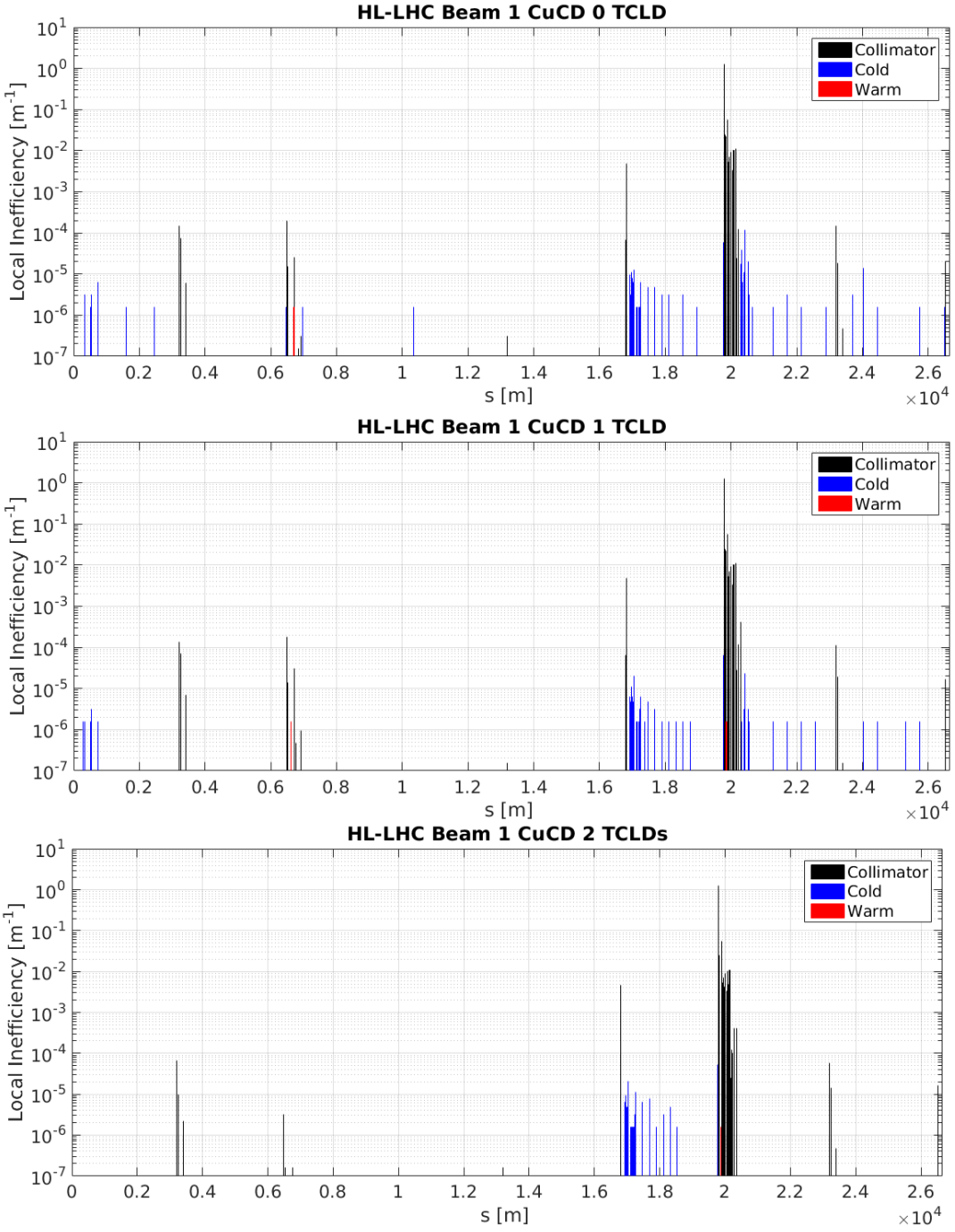

Figure 6.31: HL-LHC loss maps with CuCD for secondary collimator jaws, comparing the use of 0, 1, and 2 TCLDs. There is a clear reduction in cold losses throughout the machine, however momentum collimation hierarchy violation is also observed when using two TCLDs.

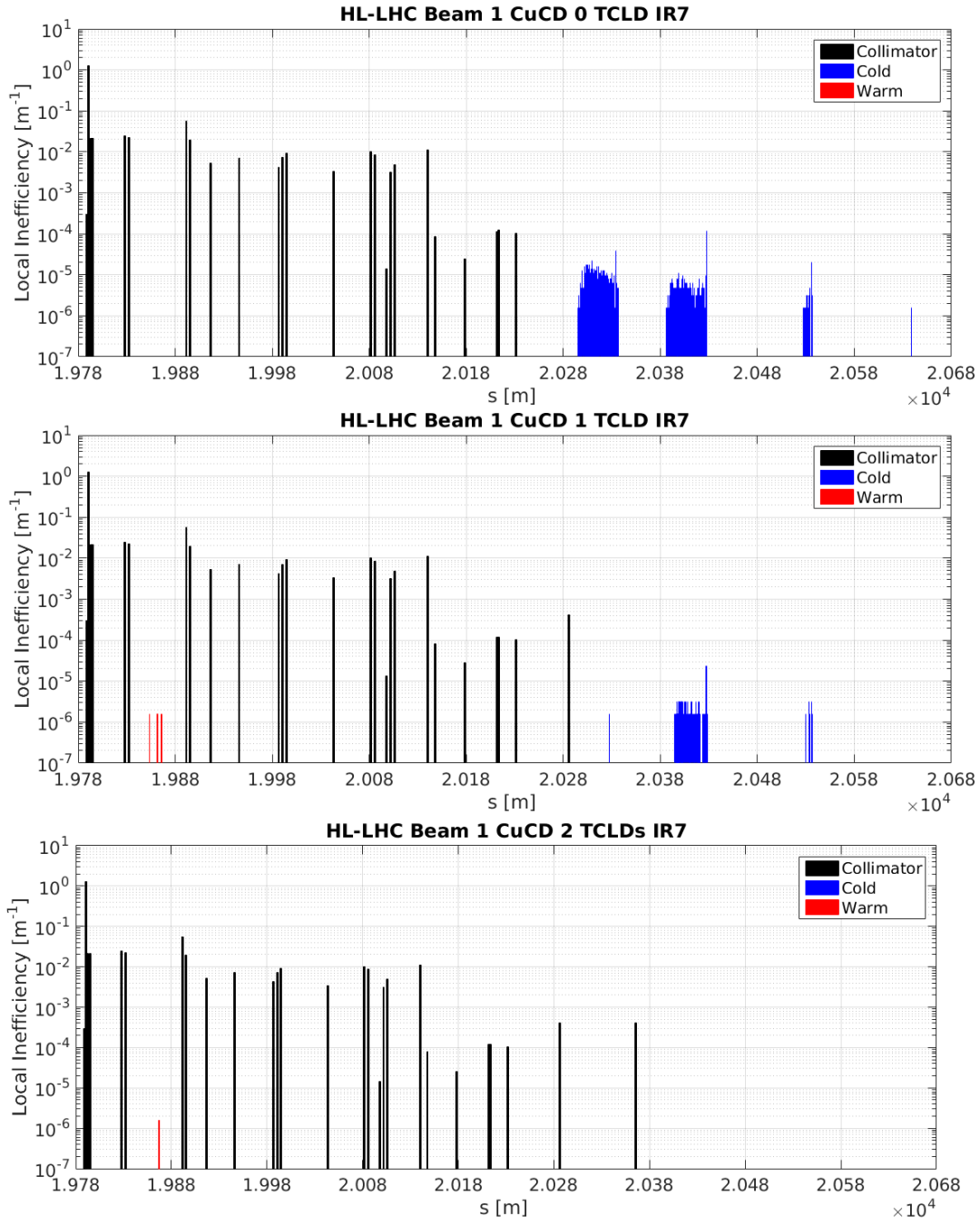

Figure 6.32: HL-LHC loss maps with CuCD for secondary collimator jaws, comparing the use of 0, 1, and 2 TCLDs, showing IR7. We notice that all cold losses in the dispersion suppressor are mitigated with the use of two TCLDs and CuCD secondary collimators.

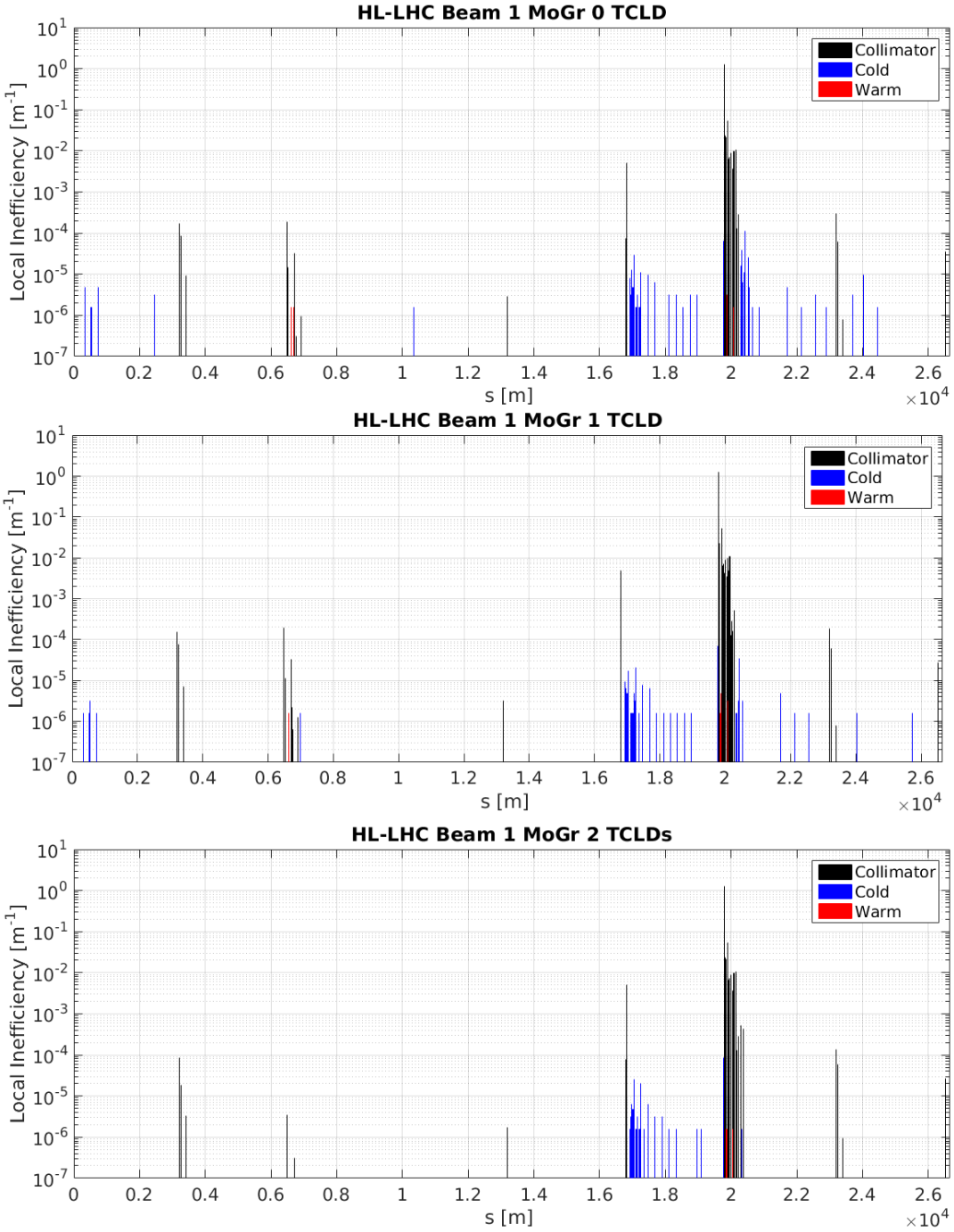

Figure 6.33: HL-LHC loss maps with MoGr for secondary collimator jaws, comparing the use of 0, 1, and 2 TCLDs. Again there is a clear reduction in cold losses throughout the machine, however momentum collimation hierarchy violation is observed when using two TCLDs.

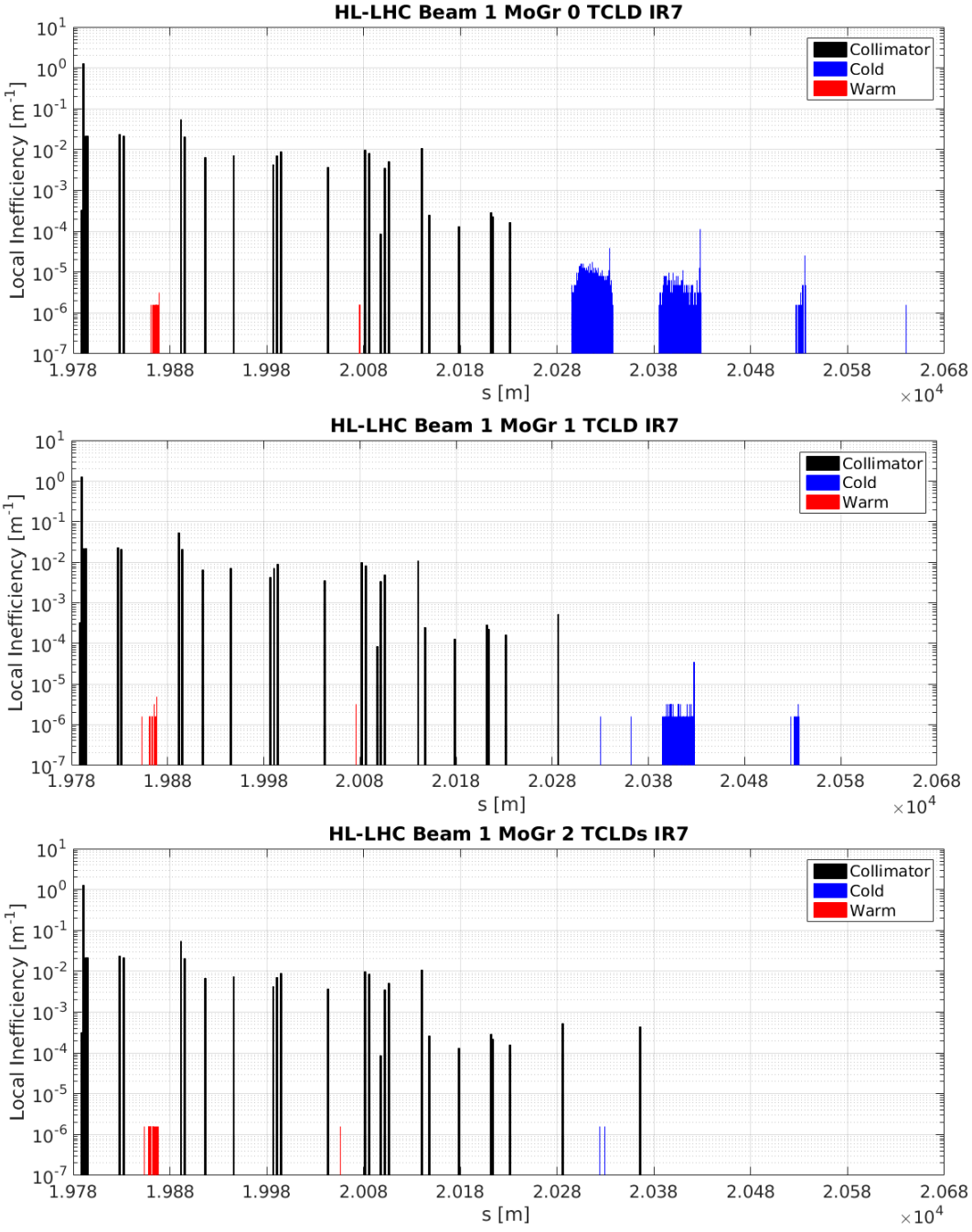

Figure 6.34: HL-LHC loss maps with MoGr for secondary collimator jaws, comparing the use of 0, 1, and 2 TCLDs, showing IR7. We see that nearly all cold losses in the dispersion suppressor are mitigated with the use of two TCLDs and MoGr secondary collimators.

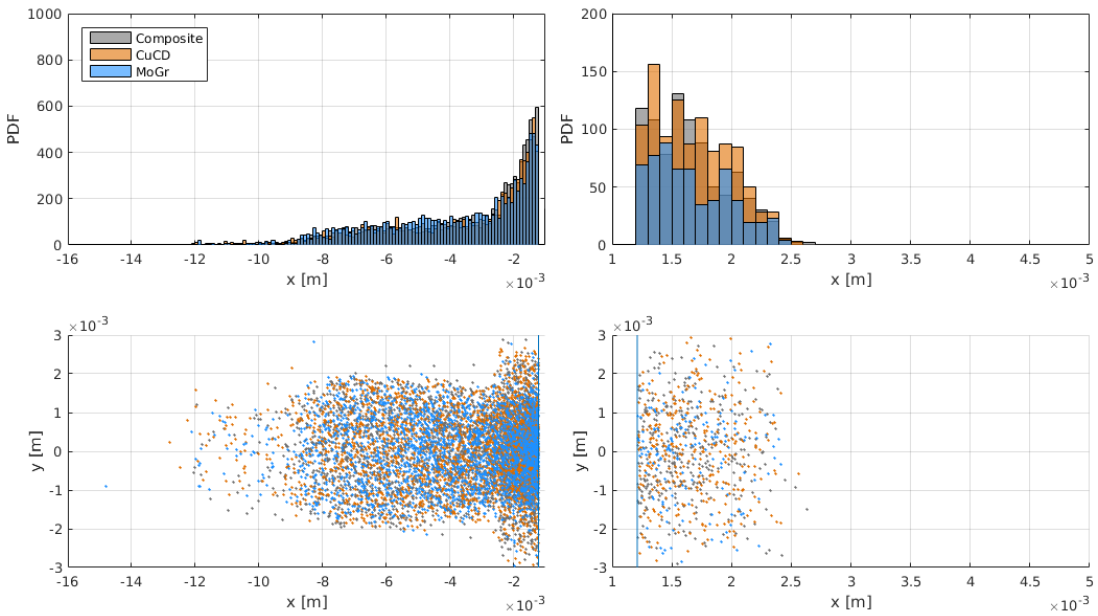

Figure 6.35: Jaw impact distribution for TCLD.8R7 comparing the effect of CFC (labelled Composite), CuCD, and MoGr secondary collimator materials. The TCLD collimator is always constructed from IMET180. The losses are nearly all on the inside jaw as this collimator is dominated by protons that have undergone a single diffractive interaction in the previous collimators.

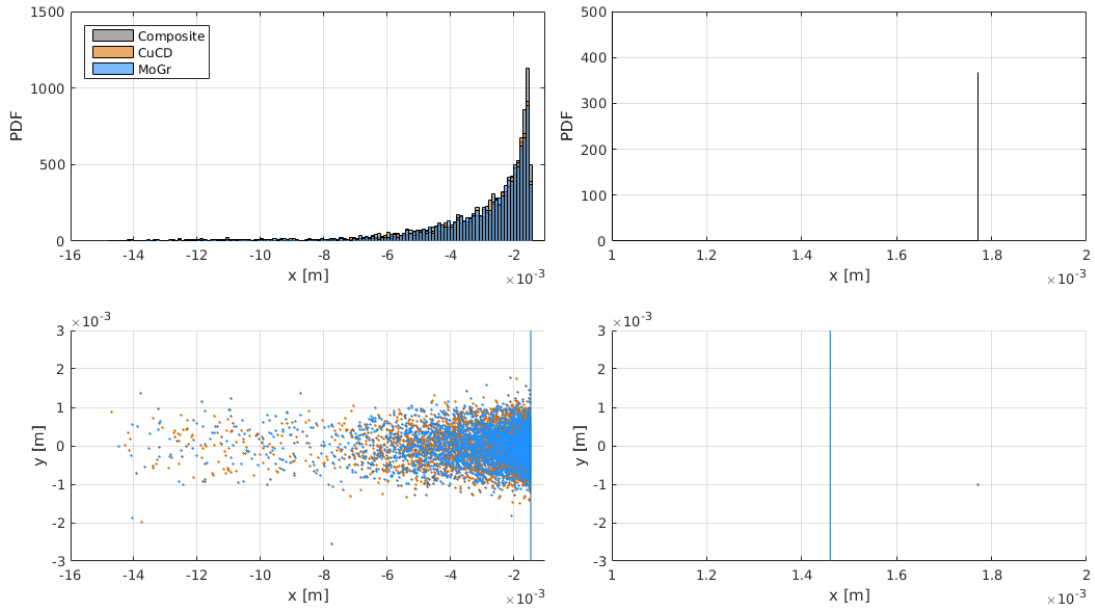

Figure 6.36: Jaw impact distribution for TCLD.10R7 comparing the effect of CFC (labelled Composite), CuCD, and MoGr secondary collimator materials. The TCLD collimators are always constructed from IMET180. In this case the losses are all (bar one) on the inside jaw as this collimator is dominated by protons that have undergone a single diffractive interaction in the previous collimators.

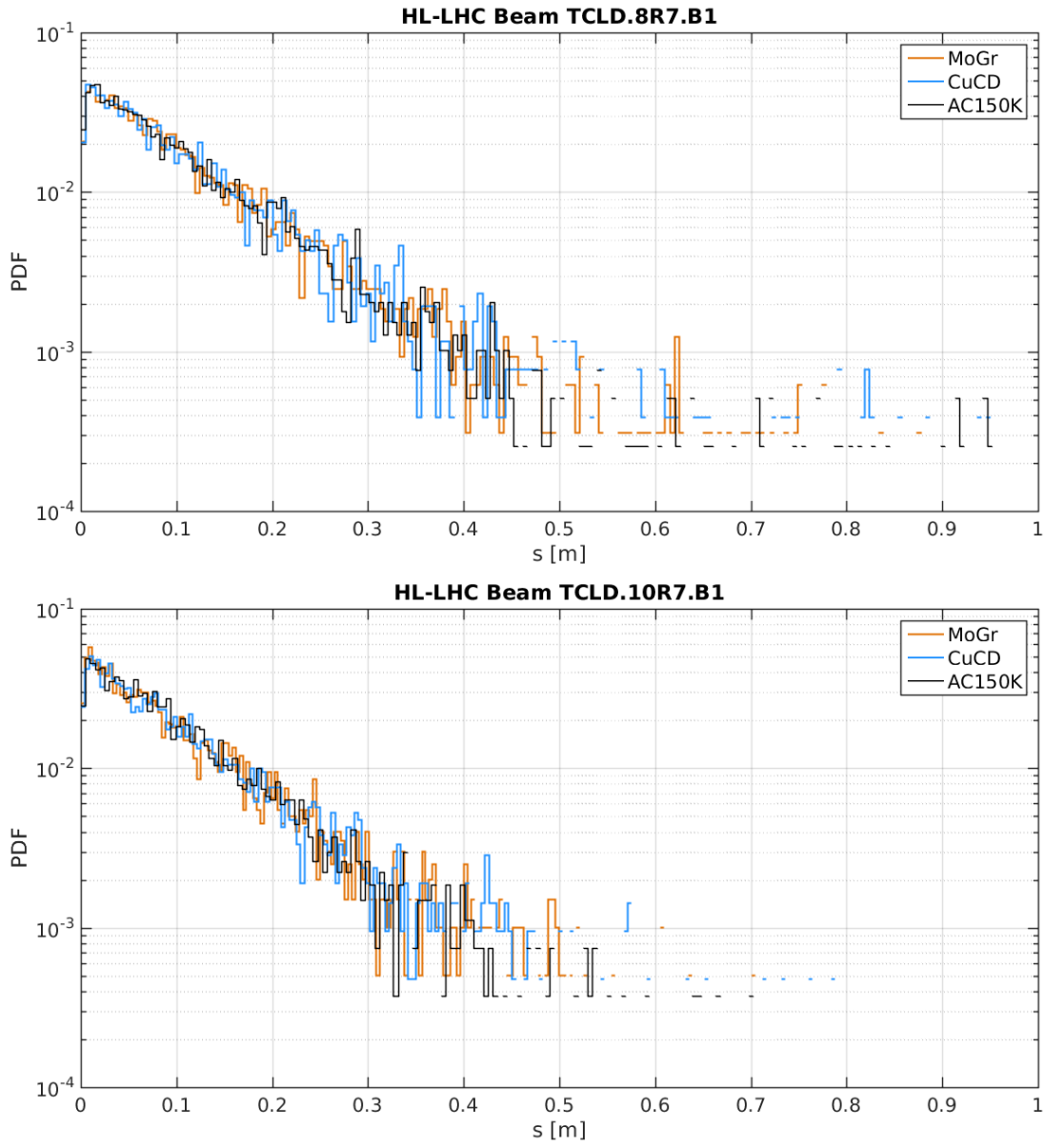

Figure 6.37: Loss distribution for TCLD.8R7 and TCLD.10R7 comparing the effect of CFC AC150K, CuCD, and MoGr secondary collimator materials. The TCLD collimator is always constructed from IMET180. This indicates that a length of 0.6 m may be sufficient for these collimators.

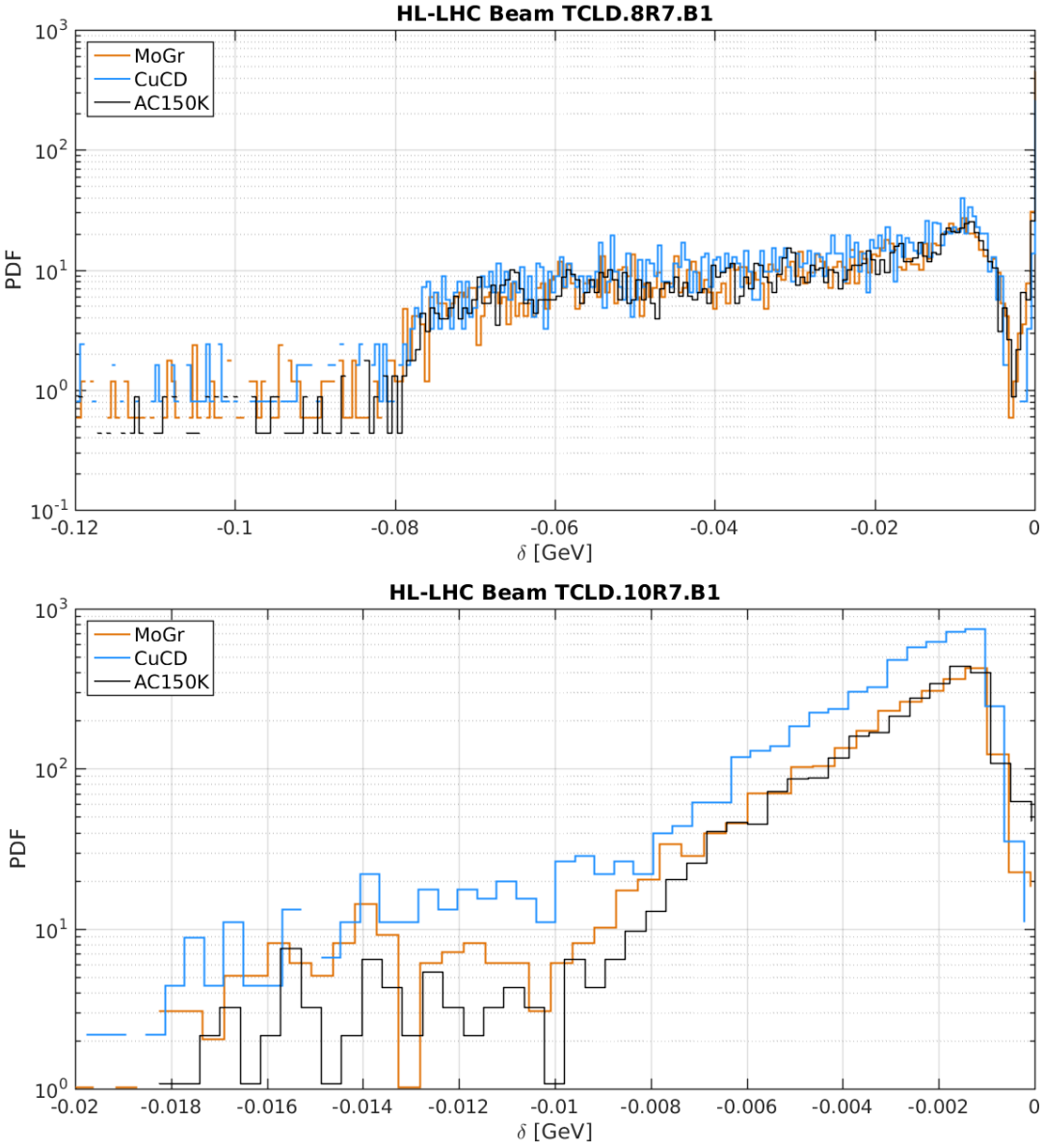

Figure 6.38: δ distribution of lost protons in TCLD.8R7 and TCLD.10R7 comparing the effect of CFC AC150K, CuCD, and MoGr secondary collimator materials. The TCLD collimator is always constructed from IMET180.

6.1.3 Summary

Using the merged and updated version of MERLIN, we have observed the effect of using composite materials over pure elements when simulating proton scattering at different energies in collimator jaws. We have observed the differences in losses summarised in Table 6.12. Evidently the difference is significant enough to warrant the improved composite method over the pure element approximation.

Region	6.5 $[TeV]$ **Beam 2**	**Nominal Beam 1**	**HL-LHC Beam 1**
Total	-0.045%	-0.65%	-0.35%
IR7 Primaries	-4.3%	-4.4%	-3.9%
IR7 Secondaries	+26%	+18.5%	+16.4%
IR7 Absorbers	+132%	+93.7%	+80%
IR7 Cold	+31.1%	+25.8%	+18.9%

Table 6.12: Percentage change in losses when switching pure elements with standard composite materials for collimator jaws.

In the HL-LHC case the addition of two TCLDs in IR7 after the betatron collimation insertion result in a drastic reduction of cold DS losses as expected. Though the halo leakage from the collimation insertion is desirable, there is indication of a violation of momentum cleaning hierarchy which is a major concern. Not only could this interfere with the efficiency of the collimation system, but it could cause a greater cold loss in the IR7 DS region than it was designed to remove.

When using novel composite materials for primary collimators in the nominal LHC a reduction in cold DS losses of 48% and 22% is given for CuCD and MoGr respectively. The use of novel materials alone is not enough to remove the DS losses, and due to other requirements such as robustness, replacing primary collimator materials is not likely to occur for the HL-LHC upgrade. In the HL-LHC, the two novel materials have a similar effect on loss maps, though as CuCD has a much smaller path length more protons are lost in the betatron collimation insertion when using CuCD secondaries. Both materials offer an improvement on CFC AC150K in terms of reduction in halo leakage from IR7.

Combining novel secondary collimator materials with TCLDs offers some insight into the likely collimation system for the HL-LHC. CuCD and MoGr offer viable alternatives to CFC AC150K in terms of proton scattering and collimation efficiency, however there are other material considerations that are currently being investigated, such as radiation hardness. TCLDs are desirable however further investigation into their effect on momentum collimation is required. For the first time we have investi-

gated the use of TCLDs in the HL-LHC using MERLIN, as well as combining them with novel secondary collimator composite materials, demonstrating the versatility of MERLIN.

Insertion Region	Collimator Family	Setting [σ]
7	Primary	6
	Secondary	7
	Absorber	10
3	Primary	15
	Secondary	18
	Absorber	20
1	Tertiary	8.3
	Absorber	10
5	Tertiary	8.3
	Absorber	10
6	Dump Protection	8
	Secondary	7.5
2	Tertiary	8.3
8	Tertiary	8.3

Table 6.13: HL-LHC collimation settings used for HEL simulations.

6.2 HEL

Here we consider the use of a HEL as a collimation enhancer, investigating the possible HL-LHC integration point. As discussed in Chapter 5 we consider three HEL integration points; the round (IP4 - 30 m), non-round (IP4 - 88.6 m), and oval beam positions (IP4 - 119 m). No TCLDs are present in these simulations as they were performed with lattice version 1.2.0 with collision optics and no thick lens TCLD collimators implemented. The collimator settings used are shown in Table 6.13, and full collimator scattering is performed. The HEL uses the LHC parameters; a current of 5 A, electron beam energy of 10 KeV, and active length of 3 m.

An initial HEL halo bunch populated between $4 - 5.8$ σ is used in the following simulations with no longitudinal component. Particle survival is shown in this section, this is defined as the number of particles that remain of the original halo bunch, as a function of turn. Note that the bunch is constructed such that at the round integration point the entire halo interacts with the HEL, however this is not true at the non-round and oval positions. 5.8 σ is chosen as the primary betatron collimator setting is 6 σ, thus any particle impacting upon this collimator must have been excited onto it via the HEL. Simulations are run for 10^5 turns, which is ≈ 9 s of operation.

6.2.1 Current Modulating Operation Modes

We begin by comparing the current modulating operation modes at the three integration points. The following figures display particle survival as a function of turns in the machine. Survival when using a HEL at the round position is shown in Fig. 6.39, at the non-round position in Fig. 6.40, and at the oval position in Fig. 6.41.

As observed in Chapter 5 the diffusive mode offers the greatest collimation enhancement. The AC mode (which has not been optimised in terms of defining parameters) only kicks those particles nearest to the collimator on to it, performing an almost instant cut of these halo protons. We confirm that, as expected, the round position offers greatest enhancement, with a 54% halo removal after 10^5 turns, this is around 21% at the non-round, and 18% at the oval positions.

It is interesting to note that at the non-round and oval positions the AC mode offers comparable cleaning enhancement to the diffusive mode after 10^5 turns, though it is clear that the diffusive mode would cause greater enhancement over a longer period of time whereas the AC mode appears to plateau.

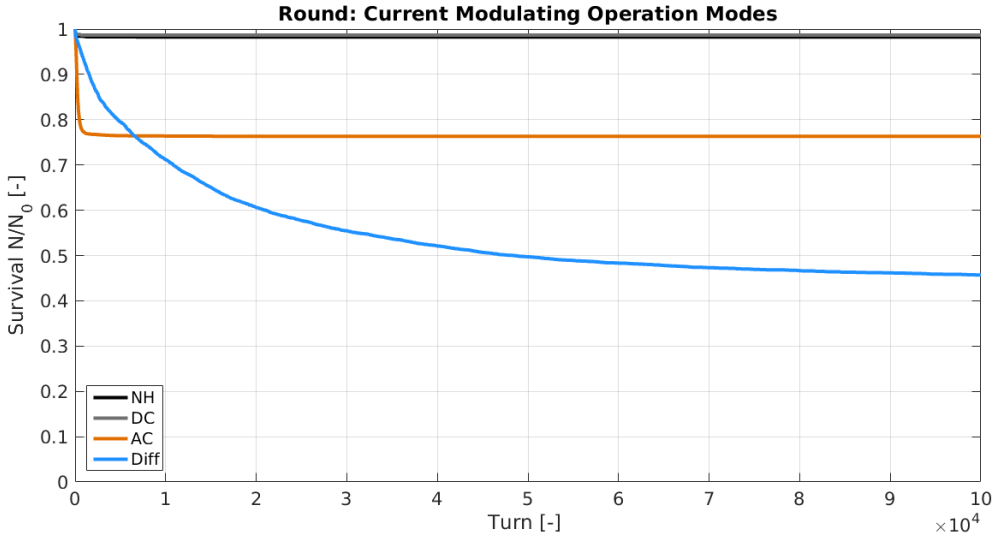

Figure 6.39: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC with no HEL (NH), and DC, AC, and diffusive current modulating HEL operation modes, at the round beam position (s = IP4 - 30 m). The diffusive mode offers continuous halo diffusion whereas the (unoptimised) AC mode gives an almost immediate but short enhancement as seen in Chapter 5.

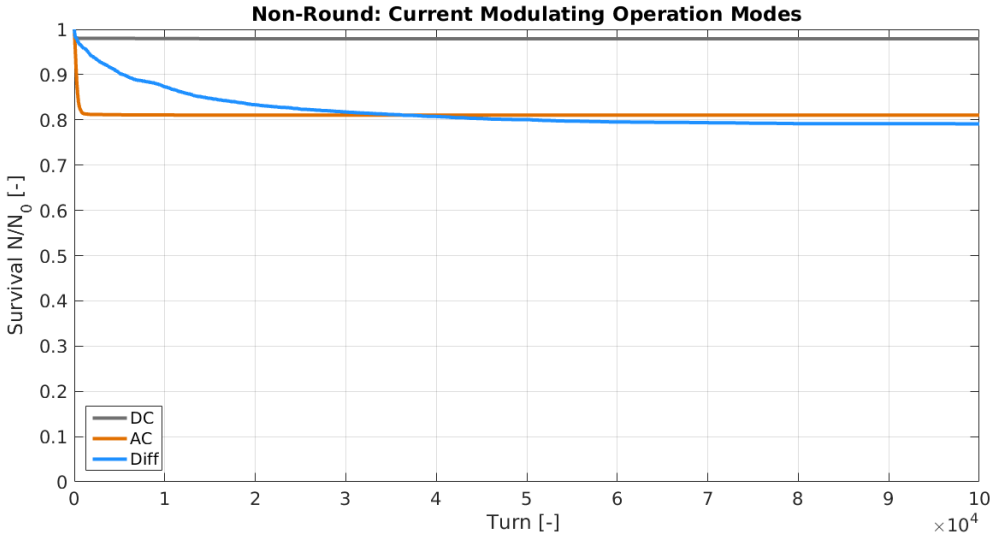

Figure 6.40: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC with no HEL (NH), and DC, AC, and diffusive current modulating HEL operation modes, at the non-round beam position (s = IP4 - 88.6 m). Cleaning enhancement for all modes is reduced as compared to using the same HEL at the round position.

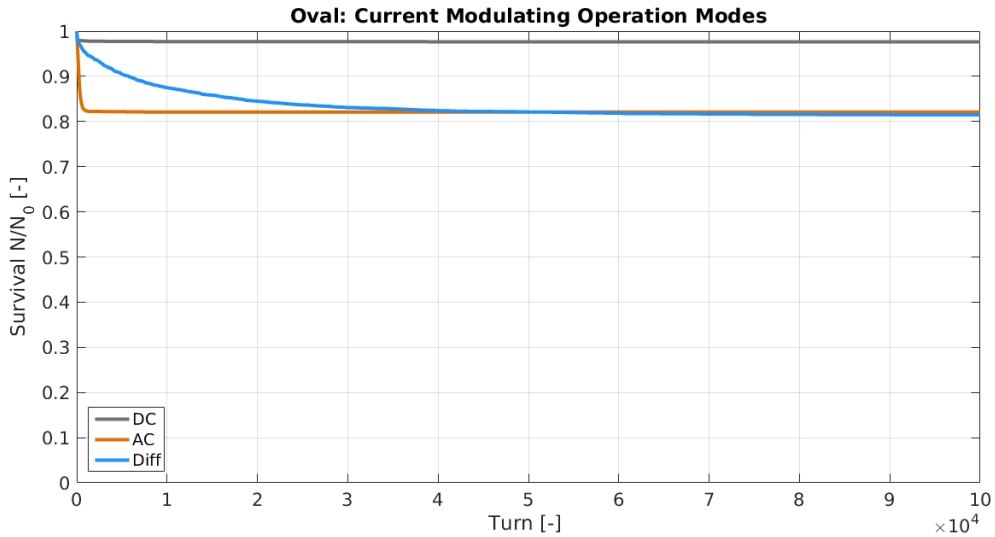

Figure 6.41: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC with no HEL (NH), and DC, AC, and diffusive current modulating HEL operation modes, at the oval beam position (s = IP4 - 119 m). Cleaning enhancement for all modes is reduced as compared to using the same HEL at the non-round position.

6.2.1.1 DC

As seen from Figures 6.39 - 6.41, the DC mode offers almost no collimation enhancement in the HL-LHC, with a halo reduction of 1.3%, 1.98%, and 2.24% at the round, non-round, and oval positions respectively. It shall therefore be ignored for the remainder of this study. This is likely due to the optics being used, which has low chromaticity and octupole current. Octupoles provide resonances which can be exploited by the HEL, and particles are more likely to experience such resonances with increased chromaticity.

6.2.1.2 AC

There is a small improvement at round position when using the AC mode (not optimised), though collimation enhancement between the non-round and oval positions are similar, as shown in Fig. 6.42. The halo is reduced by 23.7%, 18.84%, and 17.89% at the round, non-round, and oval positions respectively.

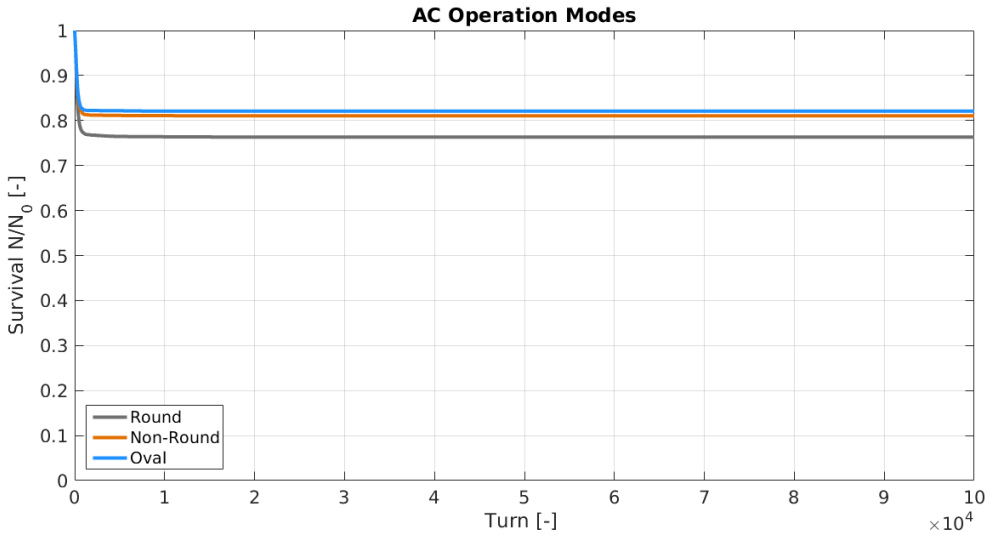

Figure 6.42: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in AC mode, at the round, non-round, and oval beam positions. The immediate enhancement is large, indicating that optimisation of this operation mode may offer a collimation enhancement similar to the diffusive operation mode.

6.2.1.3 Diffusive

We clarify the significant improvement at the round position when using the diffusive mode in Fig. 6.43. We aim to bring the non-round collimation enhancement closer to that of the round position, which is the motivation for attempting novel HEL operation modes.

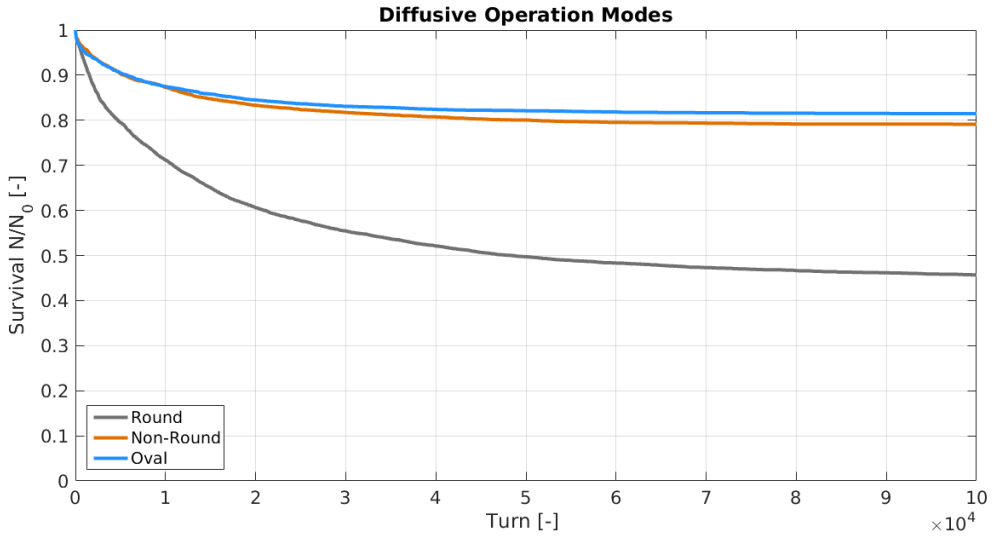

Figure 6.43: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode, at the round, non-round, and oval beam positions. As expected this mode gives a continuous collimation enhancement.

Halo survival for the existing HEL operation modes in the HL-LHC are summarised in Table 6.14.

Operation Mode	Round	Non-Round	Oval
DC	98.7	98.02	97.76
AC	76.3	81.16	82.11
Diffusive	45.69	79.15	81.47

Table 6.14: Particle survival $\frac{n}{n_o}\%$ for an initial halo of 10^4 particles between 4 - 5.8 σ after 10^5 turns in the HL-LHC with a HEL in the corresponding position using existing operation modes.

6.2.1.4 Collimation Enhancement

In practical operation the AC mode requires good knowledge of the machine tune, and numerical simulations have shown that a number of AC parameters (see Chapter 4) must be optimised for each machine lattice. The diffusive mode offers much simpler operation as either the HEL may be switched on or off or the current may be modulated randomly, on a random turn-by-turn basis. Though this may appear to be a boon, it may not be practical for the cathode to operate in the diffusive mode when compared to the AC, as it is more demanding to switch the cathode and corresponding electronics and magnets of the HEL continuously from zero to maximum current. The AC mode allows comparatively gentle ramping of currents when the parameters are set.

As no investigation of AC parameters was performed the AC mode in these simulations may not be enhancing collimation as well is possible. This provides another direction for future investigation using MERLIN.

6.2.2 Geometrical Operation Modes

Assuming the diffusive mode of operation to be the most effective in enhancing collimation, we propose a number of novel geometric and dynamic operation modes designed to improve the cleaning enhancement at non-round beam HEL positions. These novel modes all use the diffusive method of current modulation.

6.2.2.1 Elliptical Matching

The first novel operation mode is the elliptical matching algorithm defined in Chapter 3. Figure 6.44 compares this mode at the non-round and oval positions with the diffusive modes at all three integration positions. This method appears to reduce the collimation enhancement, which is not surprising as the non-round and oval positions have beams that are larger in x than y. This means that the elliptical matching attempts to reconcile coverage of the vertical halo, which reduces the enhancement in the horizontal plane. It is interesting to see that in the non-round position the survival is similar after 10^5 turns, attempting simulations with many more turns could indicate an improvement in collimation enhancement over a longer time period.

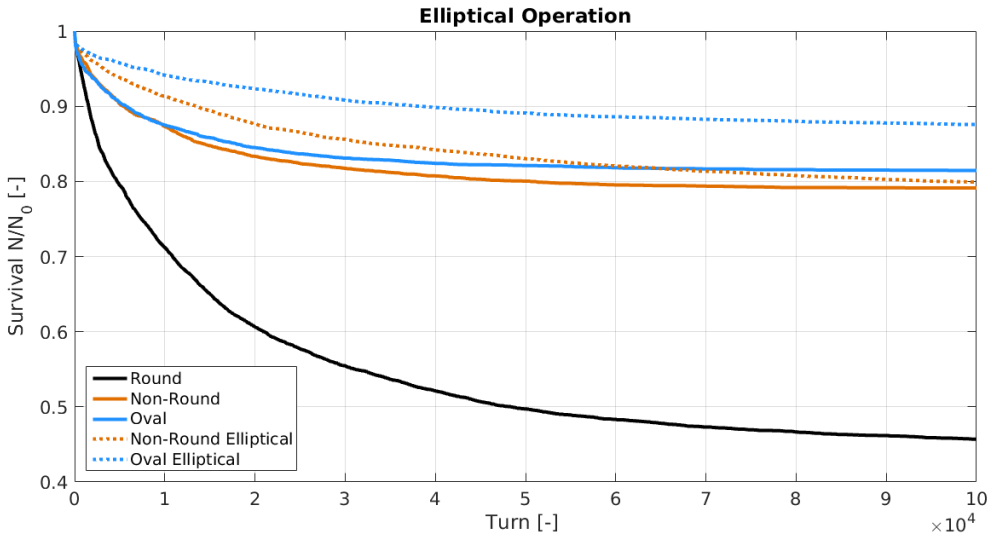

Figure 6.44: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with elliptical matching, at the non-round, and oval beam positions. For comparison the survival for a HEL in diffusive mode with no geometrical enhancement at the round, non-round, and oval beam positions are also shown. The elliptical matching offers no improvement at the non-round and oval positions.

6.2.2.2 Pogo Operation

As the elliptical matching attempts to cover one side of the elliptical halo (in this case the positive vertical), the dynamic Pogo mode alternates between both semi-minor sides of the halo (in this case positive and negative vertical). Halo survival is shown in Table 6.15 and compared with normal diffusive operation, we observe that this first attempt at dynamic operation offers similar collimation enhancement to the static elliptical matching. Particles in the beam halo do not maintain their position each turn, they undergo betatron and synchrotron oscillations that cause them to trace an elliptical Poincaré section at the HEL. This means that as well as the particle movement, the HEL is randomly switched on and off, and the HEL may or may not cover the particle in question as it is being translated vertically back and forth. It is likely the vertical translation negates any possible improvement due to increased halo coverage.

Operation Mode	**Non-Round**	**Oval**
Diffusive	79.15	81.47
Elliptical	79.9	87.61
Pogo	79.76	88.2
Hula	85.37	90.52
Close Hula	78.18	82.25

Table 6.15: Particle survival $\frac{n}{n_o}\%$ for an initial halo of 10^4 particles between 4 - 5.8 σ after 10^5 turns in the HL-LHC with a HEL in the corresponding position and operation mode. The aim is to approach the collimation enhancement shown at the round HEL position, which is 45.69%, it is clear that these dynamic modes do not rectify the decreased halo removal due to the beam not being round.

6.2.2.3 Hula Operation

The major concern when using the dynamic Pogo HEL operation is that the HEL does not overlap sufficiently with the horizontal halo as it attempts to cover the vertical halo. In order to remedy this another dynamic operation mode, the Hula mode, translates the elliptically matched HEL between the four horizontal and vertical maxima of the transverse beam. The result of this operation mode in the HL-LHC is shown in Table 6.15, we observe that the Hula operation reduces the collimation enhancement. This is likely due to the fact that when the HEL is translated to touch the horizontal extremities of the bunch core, the minimum radius is too large due to the elliptical matching, and halo coverage is reduced.

6.2.2.4 Close Hula Operation

In order to improve upon the Hula operation, the Close Hula operation mode maintains the minimum HEL radius as well as translating it around the beam core. We observe from Fig. 6.45 that this method offers a very small improvement on cleaning enhancement after 10^5 turns at the non-round position, though this is not enough to approach that of the round position. As summarised in Table 6.15 the Close Hula operation reduces the halo to 78.18% and 82.25% at the non-round and oval positions respectively after 10^5 turns.

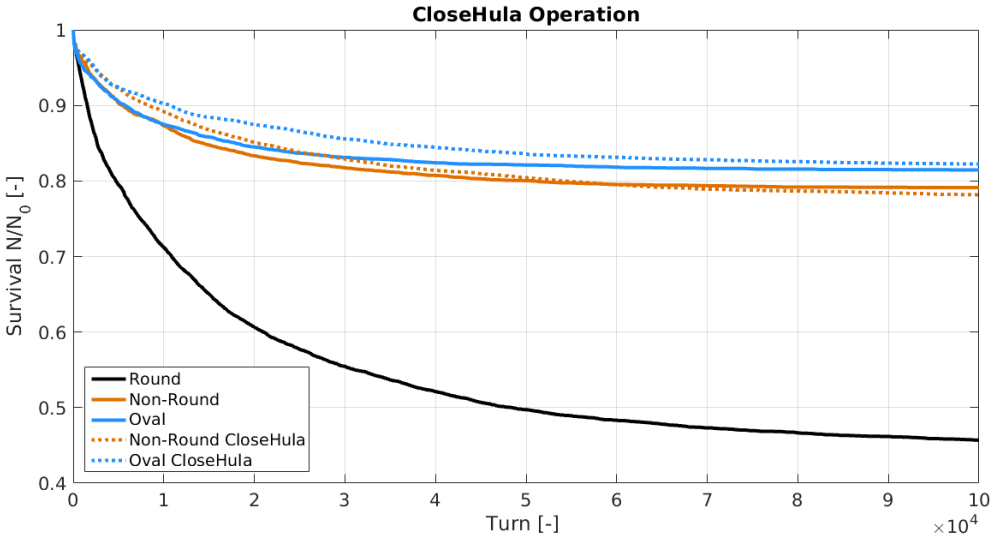

Figure 6.45: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with dynamic Close Hula operation, at the non-round, and oval beam positions. For comparison the survival for a HEL in diffusive mode with no geometrical enhancement at the round, non-round, and oval beam positions are also shown. This dynamic mode offers a very small improvement at the non-round position, but not at the oval position.

6.2.3 HEL Beam Properties

As the dynamic operation modes offer no significant cleaning enhancement improvement at the non-round and oval positions, we perform a brief investigation of the main HEL parameters: the active length, electron current, and electron energy. Increasing these parameters may be preferable to using novel operation modes in order to improve collimation enhancement at non-round beam positions.

6.2.3.1 e^- Energy

First we increase only the electron beam energy from 10 KeV to 15 KeV and 20 KeV. The resulting particle survival at the non-round position is shown in Fig. 6.46. We observe that increasing the energy decreases the collimation enhancement. To understand this we must recall the kick given by the HEL, equation 3.7. The electron energy is present in this equation in the form of the normalised energy β_e. A β_e term is present on both the numerator and denominator, however the rigidity term in the denominator is much larger than the remaining numerator. Increasing the electron energy results in a decreased kick. In reality a sufficiently high electron energy must be used in order to reduce the force from the proton beam which could distort the electron beam greatly, thus negating its effectiveness, and likely interfering with the proton beam core. Table 6.16 confirms that this reduction occurs at all chosen integration positions, and doubling the electron beam energy results in a $\approx 20\%$ decrease halo cleaning after 10^5 turns.

6.2.3.2 e^- Current

Next we increase only the electron beam current. From the kick equation 3.7 we expect a linear increase in collimation efficiency as the current is increased. The resulting survival is shown in Fig. 6.47. This shows the expected behaviour, an increase in HEL beam current results in an increased kick and therefore improved halo removal. This is the highest achieved improvement on collimation enhancement at the non-round position with a survival of 66.01% after 10^5 turns.

Table 6.16 shows the survival at the three integration points compared to that when doubling the beam current. We observe the improvement in all cases, but note that the improvement is larger at the oval position when compared to the non-round. This implies that the position of the kick, or rather the phase advance between the HEL and the collimators, is of great importance. Drawing the conclusion that an increase in current equates to an increase in collimation enhancement up to a certain limit is

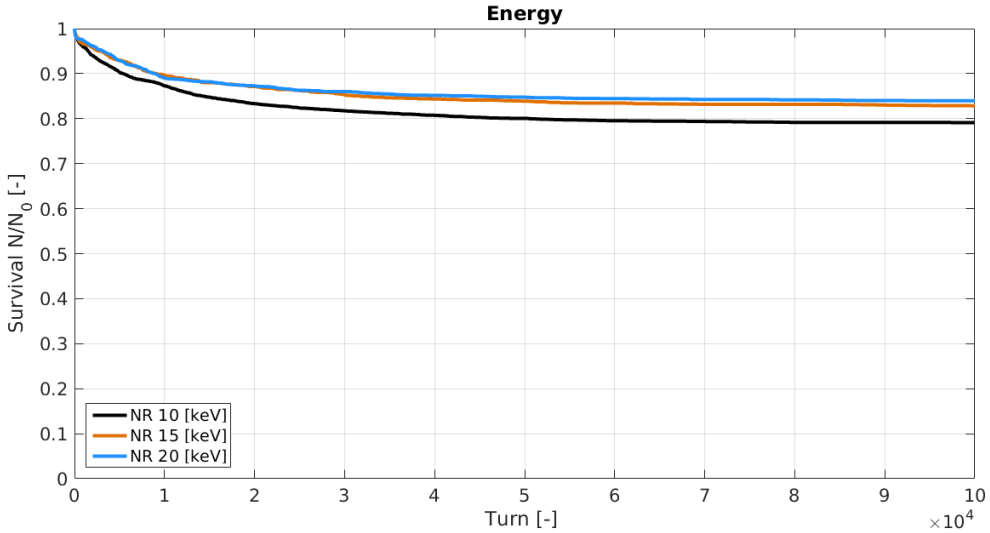

Figure 6.46: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode for various HEL energies, at the non-round position. As explained in the text increasing the HEL beam energy results in a reduced collimation enhancement.

naive. The collimation enhancement is dependent on many factors, and conclusions drawn here may only be valid for the specific simulated set up. For this reason we attempt to draw conclusions on behaviour rather than quantify possible achievable collimation enhancement.

6.2.3.3 Interaction Length

Finally we compared the effect of increasing the active length of the HEL. As expected there is a linear increase in collimation efficiency as the length is increased, as shown in Table 6.16. We don't observe the same limit as when increasing the current, however the halo cleaning does not approach the limit previously identified in the non-round position. We observe the effect of increasing the active length from 3 m to 5 m at each of the integration points in Fig. 6.48. In this case the 170% increase in length results in an increase of 146%-207% in halo cleaning.

6.2.4 Integration

We also attempt to increase all three of the HEL parameters that directly impact upon the kick, this is shown in Fig. 6.49 where a HEL current of 10 A, length of 5 m,

Operation Mode	Round	Non-Round	Oval
Diffusive	45.69	79.15	81.47
HEL Beam Energy			
10 [keV]	45.69	79.15	81.47
15 [keV]	-	82.97	-
20 [keV]	56.35	83.68	84.74
HEL Beam Current			
5 [A]	45.69	79.15	81.47
6 [A]	-	76.14	-
7 [A]	-	72.06	-
8 [A]	-	71.35	-
10 [A]	26.09	66.01	76.12
HEL Active Length			
3 [m]	45.69	79.15	81.47
4 [m]	-	73.8	-
5 [m]	29.83	69.42	73.98
5 [m] 10 [A] 20 [keV]	21.58	61.23	68.33

Table 6.16: Particle survival $\frac{n}{n_o}\%$ for an initial halo of 10^4 particles between 4 - 5.8 σ after 10^5 turns in the HL-LHC with a HEL in the corresponding position and operation mode.

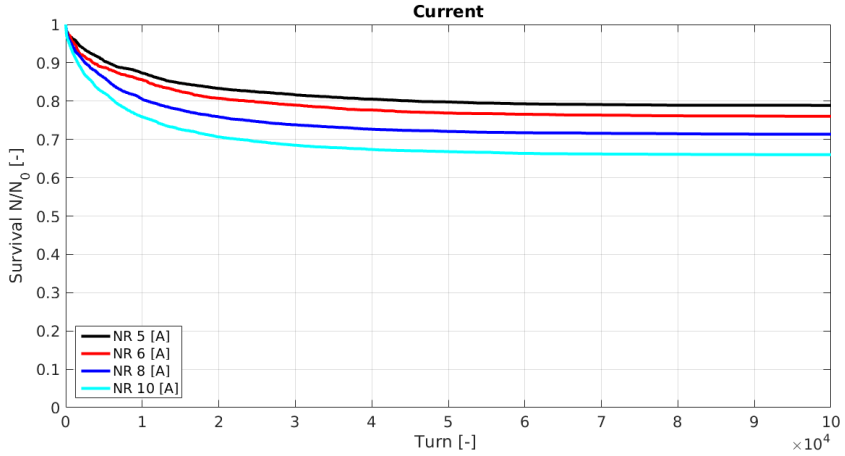

Figure 6.47: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode for various HEL currents, at the non-round position. As expected, increasing the HEL current results in an increased collimation enhancement.

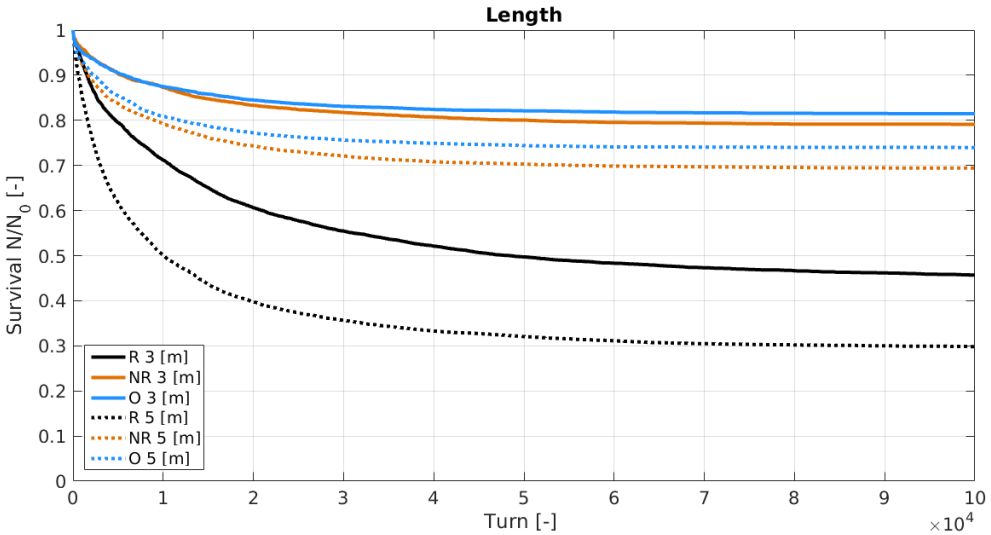

Figure 6.48: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with a length of 5 m at all three HEL positions. For comparison the standard setting of 3 m is shown for all three positions. As expected, increasing the HEL active length results in an increased collimation enhancement.

and energy of 20 keV is used at all three integration points. We observe that particle survival is halved after 10^5 turns at the round position, whereas the reduction is around 20% and 15% at the non-round and oval positions respectively. In hindsight increasing the HEL beam energy was unwise due to the results shown in Section 6.2.3.1, however these simulations were run in parallel. This shows that any improvement is reduced because of the reduced HEL overlap with the halo at non-round positions.

6.2.5 Summary

We have observed the expected reduction in collimation enhancement when integrating the HEL at a non-round beam position as compared to a round beam position. Collimation enhancement at the oval beam position is further reduced, confirming our initial hypothesis that this enhancement reduces as the beam becomes less round.

Upon comparing the current modulating operation modes we see that the DC mode offers no noticeable collimation enhancement, the AC mode causes an almost instant cut in the halo, and the diffusive mode causes a continuous diffusion of halo particles onto the collimators. This results in 54% halo removal after 10^5 turns at the round, 21% at the non-round, and 18% at the oval positions.

Whereas the diffusive mode appears to be simple as it does not require good knowl-

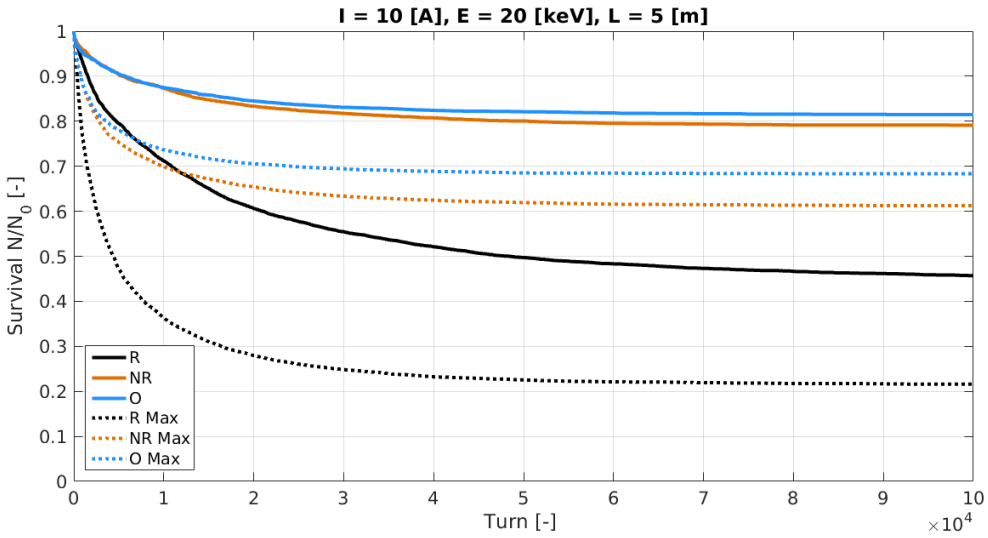

Figure 6.49: Survival fraction of a HELHalo bunch populated between 4 - 5.8 σ for 10^5 turns in the HL-LHC using a HEL in diffusive mode with a current of 10 A, length of 5 m, and energy of 20 keV at all three HEL positions. For comparison the standard setting of 5 A, 3 m, and 10 keV is shown for all three positions. At the non-round position we begin to approach the cleaning enhancement of that at the round position.

edge of the tune, from a hardware perspective random modulation of the HEL current requires rapid rise times in power supplies and the cathode, which may be very demanding. The AC mode may be preferable as it uses a relatively gentle modulation. Though the diffusive mode offers the greatest collimation enhancement, note that the AC mode has not been optimised. The requirement for AC optimisation makes the diffusive mode the obvious choice for an investigation of novel geometrical operation.

Novel elliptical matching is used to increase coverage of the halo when the beam is not round, and novel dynamic operation modes have been combined with the diffusive current modulation in order to attempt an increase in collimation efficiency at the non-round position. Through multiple iterations of dynamic operation the collimation efficiency was increased, however this improvement does not compare to that at the round position. Practically an elliptical matching may be possible, but the dynamic modes are less likely, as translation of the HEL on a turn-by-turn basis may be a slow process. The EM field felt by the beam core may be of concern as the HEL is offset from the centre of the beam, this could amplify the effect of any instabilities in the HEL beam.

It is preferable to optimise the HEL active length or current, which are proportional

to the kick, rather than use dynamic or geometric operation. When the kick is increased we improve the collimation enhancement up to a certain limit. It is postulated that this limit is the point at which the kick focusses the betatron halo rather than exciting betatron oscillations and thus diffusion. When increasing the HEL beam energy it is observed that the collimation enhancement is reduced, this is due to the kick decreasing.

As the HEL intercepts the beam halo at 4 σ the question of betatron collimation hierarchy violation arises. Whether the presence of the HEL imposes restrictions on collimator jaw openings has not been investigated in this thesis, and provides an interesting opportunity for future study.

The dynamics of the proton halo influences the HEL performance, and can affect beam lifetime, emittance growth, and collimation efficiency. Many factors influence diffusion of halo particles onto collimators or aperture restrictions. HELs are the most flexible and demonstrated tools for active halo control when using high power beams [100]. A machine development campaign has begun to measure halo diffusion rates in the LHC using collimator scans [101]. MERLIN may be used to approximate the diffusion rate given by the HEL by simulating a delta function (or pencil beam) at a given x (that falls within the HEL radii), and measuring the resulting Gaussian distribution in x after a given number of turns. Using an identical simulation in SixTrack as a benchmark, this could be useful for the HL-LHC. A required removal rate could be set using estimates for asynchronous beam dump or crab cavity failure scenarios, and the predicted HEL removal rate compared.

7

Conclusion

In 2019 the LHC as we know it will cease to operate; the machine will be upgraded to meet the demands of the High Luminosity project, which aims to rapidly increase the amount of particle physics data taken by the LHC. In order to deal with this upgrade, new materials and novel collimation schemes must be investigated. With this in mind, MERLIN has been developed as a collimation tool, and has been used to investigate the novel materials CuCD and MoGr, as well as the novel HEL collimation scheme.

MERLIN is a C++ accelerator library that, due to its modular nature, is easy to use and modify. It has been significantly upgraded since it was given to the current collaboration by Nick Walker. New algorithms include those needed for handling relevant collimation input files, proton collimation, scattering routines, loss map output, and a robust HEL processes. This describes the modifications made for MERLIN 5-01, giving a detailed overview of modifications in Chapter 4 and Appendix D.

We have performed a number of benchmarks in order to validate MERLIN. Comparing proton scattering in MERLIN with existing FLUKA, STRUCT, and K2 data for a 50 cm long collimator test case shows excellent agreement. Using a 1 cm solid material test case we have compared individual scattering routines with SixTrack, and quantified differences due to the improved scattering in MERLIN. We have also shown that the SixTrack-like scattering routines in MERLIN are comparable with those in SixTrack, meaning we may use this scattering to imitate that in SixTrack for further comparisons.

We have modelled collimator jaws as composite materials for the first time in MERLIN, and discovered that there is a large difference in the number of scatters when compared to modelling materials as pure elements. Composite materials are available in SixTrack, however an homogeneous mixture method is used. The functionality

and proper treatment of composite materials has been added to MERLIN, and shown to differ from the homogeneous approximation used in SixTrack. The homogeneous method was also added to MERLIN as a comparison tool and shown to be similar to that SixTrack. The 1 cm test case was also used to show that when a composite is not dominated by a single element, the homogeneous approximation is not valid. Thus we can conclude that properly modelling composite materials is important for loss map production.

More detailed comparisons of composite treatment and particle scattering may be performed using particle shower codes such as FLUKA, however MERLIN aims to balance accurate modelling with performance. No secondary particles are tracked in MERLIN, and any inelastic interaction is deemed a proton loss, in order to minimise simulation time.

Loss maps have been directly compared with SixTrack and are in excellent agreement. The effects of 6D and 4D tracking on loss maps has been compared, and the `SYMPLECTIC` tracker has been debugged and used for the LHC for HEL simulations. When producing loss maps for the LHC we observe the importance of using composite materials over the pure elements that have been previously used as an approximation. When using novel composites, the greatest improvement in collimation was given when replacing primary collimators, however it is more likely that secondary collimators will be replaced, so this was also investigated. It was observed that this still provides a reduction in the halo leakage from the betatron collimation insertion. Based on these results, CuCD is preferred for upgraded secondary collimators as it offers better proton absorption than MoGr, though both offer an improvement on the currently used CFC AC150K. These conclusions must be supported with experimental data. The HiRadMat [102] (high radiation to materials) experiment at CERN hopes to impact proton bunches from the SPS (super proton synchrotron) on novel collimator jaw materials in order to gauge their robustness and response. This should indicate if the proposed materials MoGr and CuCD behave as expected.

The use of TCLD dispersion suppressor collimators post-IR7 in the HL-LHC has been investigated. It is clear that the TCLDs can almost completely remove cold DS losses after the betatron collimation insertion, however they also appear to violate the momentum collimation hierarchy when using a half-gap of 10 σ. This needs to be investigated thoroughly; will they perform the required mitigation of cold DS losses with a larger half-gap, without violating the momentum collimation hierarchy? The TCLD material may also require appraisal, as it has been shown that an asynchronous beam dump may cause catastrophic TCLD loss rates [103]. We have shown (assuming the jaw

material remains Inermet) that an active length of 0.6 m should be sufficient, which is useful as the baseline may be reduced to this value (from 1 m) due to space constraints. One could ascertain the safe half-gap of the TCLDs by performing momentum collimation simulations with MERLIN, using an initial momentum halo distribution rather than the betatron halo used in this thesis.

A HEL algorithm has been added to MERLIN and benchmarked against SixTrack. The algorithm makes a number of assumptions that could be built upon in future work. These include ignoring the regions at the start and end of the HEL where the electron beam may overlap the proton beam, and the toroidal magnetic fields used to bend the HEL beam into and out of the proton beamline. The diffusive current modulating operation mode has been identified as the most promising in this simulation campaign, though this may be due to lack of proper optimisation of the AC operation mode, which was not performed. A proper optimisation of the AC parameters should be performed for each lattice version used, it would be useful to develop an algorithm in MERLIN that automated this optimisation for future work. Though the diffusive mode appears the simplest, as it does not require a good knowledge of the machine tune and optimisation of a number of parameters as in the AC mode, it may not be the most hardware friendly as it requires fast and random current modulation.

The identified HEL integration position in the HL-LHC offers a non-round beam, and as the HEL beam is round this reduces its overlap with the proton halo. For the projected LHC HEL hardware the radial profile required an empirical update. The loss in collimation enhancement at non-round positions has been shown, and attempts to rectify this have been made using novel geometrical and dynamic operation modes. The culmination of these novel operation modes offers only a small improvement to collimation enhancement, thus an investigation into the effect of changing HEL parameters at the non-round position was performed. It was found that increasing the HEL beam energy decreases the kick, and increasing the HEL active length or current increases the kick. Increasing the kick is not necessarily a good thing due to the focussing effect of the kick. We observe an improvement to collimation enhancement when increasing the HEL active length or current. There is also an indication that the phase advance between the HEL and the TCP is important, such that the integration position may need to take this into account in order to improve collimation enhancement. Practically hardware expenses increase with the length, and more space is required in the machine, increasing the current requires novel hardware. Further investigation into the HEL parameters effect on collimation enhancement would clarify the effects and limits observed. In summary we have shown that, when using the existing baseline HEL

hardware (length of 3 m, energy of 10 keV, current of 5 A) at the non-round position, it is not possible to obtain similar cleaning enhancement to that at a round beam position by using novel dynamic HEL operation. By increasing the HEL current or active length, the reduction in collimation enhancement due to the non-round beam may be diminished. This however requires a large R&D effort to increase the HEL cathode current, or extra space in the machine. In order to use the HEL as a collimation enhancer the best option is to ensure that it operates at a round beam position. The priority for HEL collimation should therefore be obtaining a more suitable integration point in the HL-LHC.

MERLIN now provides a robust and tested platform for HL-LHC collimation studies. The MERLIN collaboration plans next to add an existing crab cavity integrator [104] (whih can cause large losses under failure scenarios) in order to further the capabilities of MERLIN for modelling HL-LHC collimation.

Appendix A

Collimation

A.1 Losses

A.1.1 Transient

For transient losses the number of protons required to induce a quench is given by:

$$n_q = \frac{\Delta Q_c}{\varepsilon}, \tag{A.1}$$

where ΔQ_c is the amount of heat per unit volume necessary to increase the temperature of the SC coils of the magnet to transition temperature T_c (where the magnet will quench), and ε is the energy per proton per unit volume deposited in the SC magnet coil.

The time needed to reach a thermal equilibrium inside the magnet coil is $\tau_{metal} = 6,3\ [ms]$, at 450 $[GeV]$ and 7 $[TeV]$ respectively. If the duration of losses $\delta t << \tau_{metal}$ the heat is concentrated around the impact point. If losses are slow enough ($\delta t \geq \tau_{metal}$) that heat diffuses in the cable but is not transferred to the surrounding cryogenic system. If the loss duration $\delta t \geq \tau_{helium}$, where $\tau_{helium} = 44,8$ ms at 450 $[GeV]$ and 7 $[TeV]$ respectively, an equilibrium temperature is reached between the super-fluid helium and the cable.

Table A.1 details the quench limits for local losses for various loss durations δt. Considering an LHC bunch of $1.15 \cdot 10^{11}$ protons, only $4 \cdot 10^{-4}\%$ of a single bunch locally deposited in an SC magnet may cause a quench at 7 TeV.

Energy $[TeV]$	**Loss Duration** δt $[ms]$	**Quench limit for local losses** [protons]
0.45	< 6 (τ_{metal})	$1 \cdot 10^9$
0.45	≥ 6 (τ_{metal})	$2.7 \cdot 10^9$
0.45	≥ 44 (τ_{helium})	$2.5 \cdot 10^9$
7	< 3 (τ_{metal})	$4.7 \cdot 10^5$
7	≥ 3 (τ_{metal})	$8.5 \cdot 10^5$
7	≥ 8 (τ_{helium})	$3.4 \cdot 10^7$

Table A.1: Number of protons required to induce SC magnet quenches for transient losses. From [32].

A.1.2 Continuous

In order to maintain the condition of $T < T_c$ in the continuous loss case, heat must be continuously removed. The energy of impacting protons is dissipated over a certain length that can be approximated with the effective length of the secondary particle showers L_{eff}. The local proton loss rate $\tilde{R_q}$ inducing a quench can then be defined as:

$$\tilde{R_q} = \frac{R_q}{L_{eff}}, \tag{A.2}$$

where R_q is the maximum allowed proton loss rate, or quench limit. From Monte Carlo simulations [32]: $L_{eff} = 1$ $[m]$ a t450 $[GeV]$, 0.7 $[m]$ at 7 $[TeV]$.

Table A.1 displays quench limits for local transient proton losses, and table A.2 displays proton loss rates in the continuous slow loss regime as a function of energy.

Energy [TeV]	Quench Limit $R_q[p \cdot s^{-1}]$	Local Quench Limit $\tilde{R_q}[p \cdot s^{-1} \cdot m^{-1}]$
0.45	$7 \cdot 10^8$	$7 \cdot 10^8$
3.5	$2.1 \cdot 10^7$	$2.4 \cdot 10^7$
4	$1.6 \cdot 10^7$	$1.9 \cdot 10^7$
7	$5.4 \cdot 10^6$	$7.8 \cdot 10^6$

Table A.2: Maximum allowed proton loss rates R_q and local loss rates $\tilde{R_q}$ for continuous slow losses on the LHC SC magnets as a function of energy. From [32]

A.2 Energy Deposition and Loss Rates

The collimation system sets the tightest restriction on the machine aperture in order to protect the machine in normal operation and in accident scenarios. The collimators

receive the highest radiation dose and are the first objects to be hit by the beam in failure scenarios. The total transverse energy density is defined as:

$$\rho_E = \frac{EN_{tot}}{2\pi\sigma_x\sigma_y}, \tag{A.3}$$

where σ_x and σ_y are the transverse beam sizes in the horizontal and vertical plane respectively, E is the proton energy, and N_{tot} is the total number of protons. Typically $\rho_E = 1GJ \cdot m^{-2}$ at the location of primary collimators (where RMS transverse beam size $\sigma_x \approx \sigma_y \approx 240\ \mu m$) at nominal intensity ($N_{tot} = 3.22 \cdot 10^{14}\ p$) and $E = 7\ TeV$. Each individual collimator can only handle a small fraction of the beam.

The maximum allowed proton loss rate R_{loss} at the collimators is calculated from the local quench limit $\tilde{R_q}$ and local cleaning inefficiency $\tilde{\eta_c}$:

$$R_{loss} = \frac{\tilde{R_q}}{\tilde{\eta_c}}. \tag{A.4}$$

This is correlated to the beam lifetime:

$$\tau \simeq \frac{N_{tot}}{R_{loss}}. \tag{A.5}$$

The loss rate peaks in the ramping phase of the LHC, where up to 5 % of the RF-uncaptured beam is estimated to be lost [32].

For continuous losses a minimum lifetime of 1 hour is specified at 450 GeV and 7 TeV. The loss rates R_{loss} and maximum power deposition P_{loss} in the collimators were calculated from equation A.5 for a given duration T, and displayed in Table A.3.

Mode	Loss Duration T $[s]$	τ $[h]$	Quench Limit R_{loss} $[p \cdot s^{-1}]$	P_{loss} $[kW]$
Injection	cont.	1	$0.8 \cdot 10^{11}$	6
Injection	10	0.1	$8.6 \cdot 10^{11}$	63
Ramp	≈1	0.006	$1.6 \cdot 10^{13}$	1200
Top Energy	cont.	1	$0.8 \cdot 10^{11}$	97
Top Energy	10	0.2	$4.3 \cdot 10^{11}$	487

Table A.3: Maximum proton loss rates and resulting power deposition for loss durations T in given lifetime τ cases. From [105].

At the full intensity of protons, a 1 hour lifetime corresponds to a loss rate of 90 billion protons a second, which equates to 0.1 MJs^{-1}, or 100 kW.

Table A.4 displays the specified beam loads for various error scenarios.

Beam Energy [TeV]	Deposited Intensity [p]	Deposited Energy [kJ]	Duration [ns]
Injection error			
0.45	$2.9 \cdot 10^{13}$	2073	6250
Asynchronous dump (all MKDs)			
0.45	$6.8 \cdot 10^{11}$	49	150
7	$4.8 \cdot 10^{11}$	538	100
Asynchronous dump (one MKD)			
0.45	$10.2 \cdot 10^{11}$	74	225
7	$9.1 \cdot 10^{11}$	1021	200

Table A.4: Beam load deposited in collimators for injection and extraction failure scenarios. From [106].

In order to operate the machine, losses in magnets must remain below the quench limit R_q. This gives us the condition that the collimation system (and MPS) must meet:

$$\frac{N_{tot}}{\tau_b} \cdot \tilde{\eta_c} < R_q. \tag{A.6}$$

Using equations A.5, A.4, and 3.2, we may rearrange equation A.6:

$$\tilde{\eta_c} < \frac{R_q}{R_{loss}}. \tag{A.7}$$

Approximating $R_q = 5{\cdot}10^6$, and $R_{loss} = 1{\cdot}10^{11}$, and in the one hour lifetime case, we can see that the required collimation inefficiency is $< 10^{-4}$.

A.3 Hardware

The LHC collimators consist of two movable jaws which are centred and aligned with respect to the beam. Each jaw is encased in a block housing, has a back stiffener for support, and a cooling circuit to allow water cooling of the jaw (which operates at room temperature). The jaw half-openings (or half-gaps) are changed though the beam phases (injection, ramp, squeeze etc). Each jaw contains blocks of different materials (dependent on collimator type) as discussed in Chapter 3. Each collimator uses water cooling and heat exchangers in the form of copper-nickel pipes. The length of collimator jaw is also dependent on the type of collimator, however most collimators include 10 cm of tapering at either end of the jaw to reduce geometrical impedance effects. All jaws are 80 mm wide and 25 mm deep along the active length [32]. Figure A.1 shows the

taper at either end of a secondary collimator.

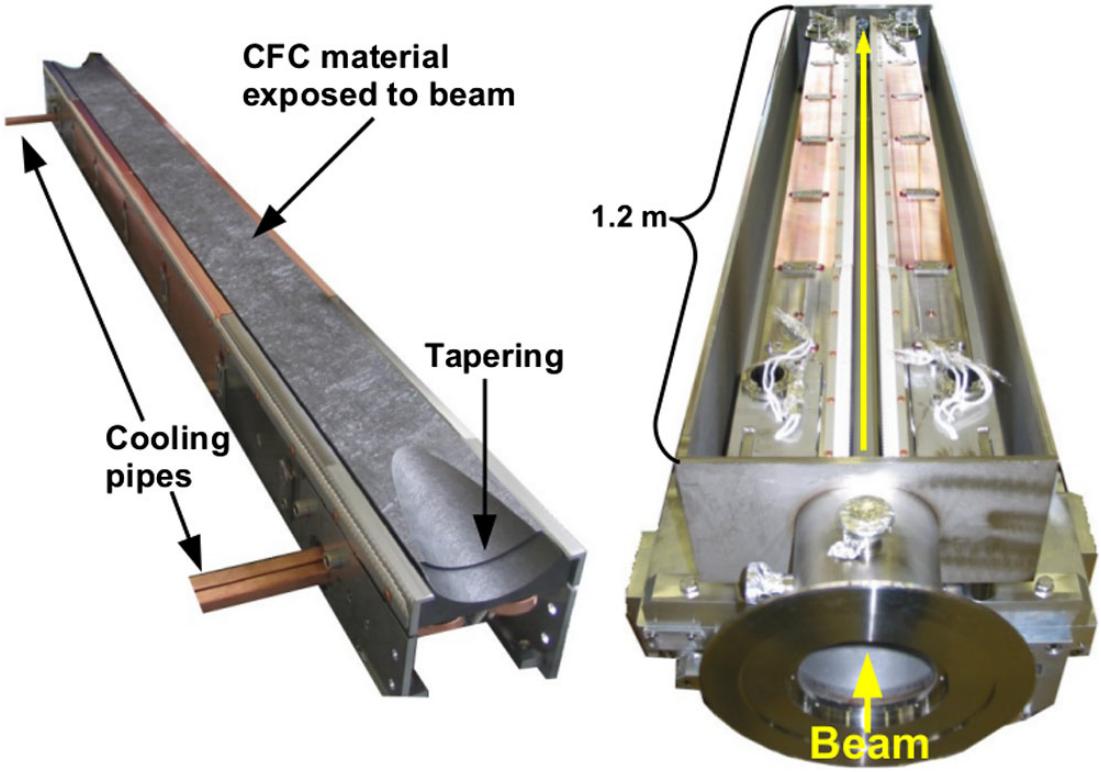

Figure A.1: Single CFC secondary collimator jaw (left) and two parallel jaws in an assembly module (right). Each jaw has 10 *cm* tapers at either end to reduce the impedance, giving the secondaries an active length of 1 *m*, and primaries 60 *cm*. From [71].

A.4 Multi Phase Collimation

A.4.1 Phase 1

Phase 1 is the central collimation system designed for maximum robustness and flexibility [107], comprised of primary and secondary collimators, and absorbers to protect sensitive machine parts.

Low Z materials were chosen such that primary and secondary collimators are highly resistant to beam impact. It is intended to continue to use the phase 1 system for the entire lifetime of the current LHC design. Phase 1 collimators are robust but impedance-limited.

A.4.2 Phase 2

Phase 2 is comprised of secondary collimators, using low impedance materials with a higher cleaning efficiency (i.e. high Z). Using higher Z material makes the phase 2 collimators less robust but increases energy deposition, thus phase 2 collimators are intended to only be used at top energy under stable conditions. Phase 2 collimators were installed in LS1.

Appendix B

Derivation of the Hollow Electron Lens Kick

First we approximate the hollow electron lens (HEL) to be an infinite line charge.

If we take an infinite line charge with charge per unit length λ, which is equal to the charge Q_e per unit length L, we may calculate the electric and magnetic fields generated by this line charge. Using the Biot-Savart law shown in equation B.1, we can calculate the magnetic flux density B at a position r from the line charge to be that shown in equation B.2. Similarly using Gauss's law shown in equation B.3 we calculate the electric field E at a position r from the line charge to be that shown in equation B.4.

$$\mathbf{B} = \frac{\mu_0}{4\pi} \cdot \int_C \frac{I \; d\mathbf{L} \times \hat{\mathbf{r}}}{r^3} \tag{B.1}$$

$$B = \frac{\mu_0 \; I}{2\pi \; r} \tag{B.2}$$

$$\oint \mathbf{E} \cdot d\mathbf{A} = \frac{Q_{enclosed}}{\epsilon_0} \tag{B.3}$$

$$E = \frac{\lambda}{2\pi\epsilon_0 \; r} \tag{B.4}$$

The Lorentz force, given in equation B.5 defines the force on a particle of charge q moving with velocity v, in an electromagnetic field. Noting that the magnetic flux density B, the electric field E, and the current I, are orthogonal, the cross product $\mathbf{v} \times \mathbf{B}$ simplifies to $vBsin(\theta)$, with $\theta = \pi/2$, thus the Lorentz force is simplified as

shown in equation B.6.

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \tag{B.5}$$

$$F = q(E \pm v_p B) \tag{B.6}$$

Here v_p is the proton velocity, we note that if the proton travels in the opposite direction to the electron, the E and B components of the force sum, and if the electron and proton travel in the same direction, the B component is subtracted from the E component.

Using the expressions for the electric field E and magnetic flux density B for an infinite line charge, and the equalities for the permeability of free space μ_0, the charge per unit length λ, the Lorentz β, the transit time t_{HEL}, and the current I, shown in equations B.7, B.8, B.9, B.10, and B.11, we may substitute into, and rearrange equation B.6, to find equation B.12.

$$\mu_0 = \frac{1}{\epsilon_0 c^2} \tag{B.7}$$

$$\lambda = \frac{Q}{L} \tag{B.8}$$

$$\beta = \frac{v}{c} \tag{B.9}$$

$$t_{HEL} = \frac{L_{HEL}}{v} \tag{B.10}$$

$$I = \frac{Q}{t_{HEL}} \tag{B.11}$$

$$F = q\left(\frac{\lambda}{2\pi\epsilon_0\ r} \pm v_p\left(\frac{\mu_0\ I}{2\pi\ r}\right)\right) = \frac{q}{2\pi r}\left(\frac{\lambda}{\epsilon_0} \pm v_p\mu_0 I\right) = \frac{q}{2\pi r\epsilon_0}\left(\frac{I}{v_e} \pm \frac{I\beta_p}{c}\right) \tag{B.12}$$

Which can once again be rearranged to find the force of the HEL on a proton at a radial distance r from the outside of the electron beam (equation B.13).

$$F(r) = \frac{Iq(1 \pm \beta_e\beta_p)}{2\pi\epsilon_0 v_e\ r} \tag{B.13}$$

As the electric field E is generated by electrons, the direction of this force on a proton is attractive, i.e. towards the propagation axis of the charge, which in this case is the centre of the HEL beam.

In order to find the angular kick given to a proton due to this force, we use the angular velocity ω shown in equation B.14, and the centrifugal force $F_{centrifugal}$ shown in equation B.15;

$$\omega = \frac{\theta}{t} = \frac{v_p}{r}, \tag{B.14}$$

$$F_{centrifugal} = \frac{\gamma m_p v_p}{r}, \tag{B.15}$$

where γ is the Lorentz gamma of the particle in the synchrotron, v its velocity, and p its momentum.

We may subsititute equations B.15 and B.10 into the angular velocity (equation B.14), to find an expression for the angular kick due to the HEL force.

$$\frac{\theta}{t} = \frac{v_p}{r} = \frac{v_p F}{\gamma m_p v_p^2}$$

$$\theta = \frac{Ft}{\gamma m_p v_p} = \frac{FL}{\gamma m_p v_p^2} = \frac{I_r q_p (1 \pm \beta_e \beta_p)}{2\pi\epsilon_0 v_e r} \frac{L}{\gamma m_p v_p^2}$$

It can be shown using equations B.5 and B.15 that the beam rigidity $B\rho$ is as shown in equation B.16;

$$B\rho = \frac{\gamma m_0 v}{q} = \frac{p}{q}. \tag{B.16}$$

Thus

$$\theta = \frac{I_r L (1 \pm \beta_e \beta_p)}{2\pi\epsilon_0} \cdot \frac{1}{v_p v_e} \cdot \frac{1}{r},$$

which gives the maximum angular kick on a proton with a normalised velocity $\frac{v}{c}$ of β_p, due to a HEL with electrons travelling at β_e, as;

$$\theta_{max}(r) = \frac{1}{4\pi\epsilon_0 c^2} \frac{2LI_r(1 + \beta_e\beta_p)}{(B\rho)_p \beta_e \beta_p} \frac{1}{R_{max}}, \tag{B.17}$$

where I_r is the charge enclosed by the radius r, and L is the length of the HEL.

We may define a function $f(r)$, which, for an ideal HEL with uniform charge density and radial symmetry, modulates the charge enclosed I_r, as shown in equation B.18.

$$f(r) = \begin{cases} 0, & r < R_{min} \\ \frac{r^2 - R_{min}^2}{R_{max}^2 - R_{min}^2}, & R_{min} < r < R_{max} \\ 1, & r > R_{max} \end{cases} \tag{B.18}$$

where:

r is the machine particle radius in xy phase space,

R_{min} is the minimum radius of the HEL in xy phase space, and

R_{max} is the maximum radius of the HEL in xy phase space.

Thus the magnitude of the HEL kick on a machine proton is defined as a function of the proton's transverse radius in xy phsae space r, the HEL current I, the HEL length L, the beam rigidity of the machine beam $(B\rho)_p$, and the HEL and machine beam normalised velocities β_e and β_p respectively, in equation B.19:

$$\theta_{kick}(r) = f(r) \cdot \frac{1}{4\pi\epsilon_0 c^2} \frac{2L_{HEL} I(1 + \beta_e\beta_p)}{(B\rho)_p \beta_e \beta_p} \frac{1}{r}. \tag{B.19}$$

Appendix C

Composite Material Properties

The following variables will be used in this appendix.

n_i atomic content (number fraction) of ith component.
m_i mass fraction of ith component.
Z atomic number.
A atomic mass.
ρ density.
σ_{el} electrical conductivity.
χ_0 radiation length.

Material	**Z**	**A** $[\frac{g}{mol}]$	ρ $[\frac{g}{cm^3}]$	σ_{el} $[\frac{MS}{m}]$	Atomic Content [%]	χ_0 $[cm]$	λ_{tot} $[cm]$
CFC	6	12.01	1.67	0.14	100 C	25.57	35.45
MoGr	6.653	13.532	2.5	1	2.7 Mo_2C, 97.3 C	11.931	24.84
CuCD	11.898	25.238	5.4	12.6	25.7 Cu, 73.3 CD, 1 B	3.162	13.56
Glidcop	28.823	63.149	8.93	53.8	99.1 Cu, 0.9 Al_2O_3	1.442	9.42
Inermet180	67.657	166.68	18	8.6	86.1 W, 9.9 Ni, 4 Cu	0.385	6.03

Table C.1: New materials in SixTrack and their corresponding atomic contents and properties [45].

Using molybdenum-carbide-graphite (MoGr) we will explain the steps required to calculate the atomic content in mass and number fraction, and thus the material properties of MoGr using the constituent elements and their respective properties. MoGr is made of molybdenum powder, graphite flakes, and carbon fibres. We begin by calculating the molecular mass of molybdenum carbide (Mo_2C), the minor component of

MoGr, using equation C.1.

$$M_{tot} = \sum_i s_i M_i, \tag{C.1}$$

where s_i is the number of atoms, and M_i is the molecular mass, of the ith element in the composite. This is simplified when a composite can be treated as a mixture of pure elements:

$$M_{tot} = \sum_i s_i A_i, \tag{C.2}$$

where A_i is the atomic mass. As Mo_2C contains 2 atoms of molybdenum and 1 atom of carbon, using eqn. C.2 we find that $M_{Mo_2C} = 203.934$. The mass fraction m_i of a constituent in a composite (in terms of molecular content) is given from equation C.3,

$$m_i = \frac{n_i M_i}{\sum_i n_i M_i}, \tag{C.3}$$

where n_i is the number fraction. This is simplified for a simple composite (treated as a mixture of pure elements):

$$m_i = \frac{n_i A_i}{\sum_i n_i A_i}, \tag{C.4}$$

Thus the mass fraction of molybdenum in Mo_2C is 0.941, or 94.1%. The atomic content (in number fractions) of MoGr is given from measurements after the production process is complete [45], and is detailed in Table C.3. Using equation C.5 [93], the mass fraction of component a in composite C may be approximated.

$$m_a = \frac{\frac{\rho_a \cdot \rho_b}{\rho_C} - \rho_a}{\rho_b - \rho_a}, \tag{C.5}$$

where a and b are the constituents of composite molecule C, and ρ is the density of the respective material. An example, for the mass fraction of Mo_2C in MoGr is given in equation C.6. This approximation is used because during the construction process the molybdenum and graphite form Mo_2C as well as a significant recrystallization of graphite [47].

$$m_{Mo_2C} = \frac{\frac{\rho_{Mo_2C} \cdot \rho_{Gr}}{\rho_{MoGr}} - \rho_{Mo_2C}}{\rho_{Gr} - \rho_{Mo_2C}}, \tag{C.6}$$

Using this equation we find the mass fraction of Mo_2C in MoGr to be 0.137 or

13.7%. We may calculate the number fraction using the atomic mass and mass fraction of each element in the composite using equation C.7:

$$n_i = \frac{\left(\frac{m_i}{A_i}\right)}{\sum_i \left(\frac{m_i}{A_i}\right)}. \tag{C.7}$$

Thus far we have obtained the number and mass fractions of Mo and C in Mo_2C, and of Mo_2C and C in MoGr. To model MoGr as the pure elements Mo and C we must use the fact that the mass fraction of molybdenum in Mo_2C is 94.1%. Of the 13.7% Mo_2C in MoGr, 94.1% is Mo and we may approximate molybdenum graphite as 12.9% molybdenum, and 87.1% carbon by mass fraction. Using equation C.7, we find the number fractions to be 1.7% Mo, 98.3% C.

A_{Mo}	95.962
A_C	12.01
n_{Mo}	$\frac{2}{3}$
n_C	$\frac{1}{3}$
m_{Mo}	0.941
m_C	0.059
ρ_{Mo_2C}	8.4 $\frac{g}{cm^3}$

Table C.2: Relative properties of a single molecule of Mo_2C.

A_{Mo}	95.962
A_C	12.01
A_{MoGr}	13.436
Z_{MoGr}	6.611
n_{Mo_2C}	0.027
n_{Gr}	0.973
n_{Mo}	0.01698
n_C	0.98302
m_{Mo_2C}	0.137
m_{Gr}	0.863
m_{Mo}	0.1213
m_C	0.8787
ρ_{Gr}	2.25 $\frac{g}{cm^3}$
ρ_{MoGr}	2.5 $\frac{g}{cm^3}$

Table C.3: Relative properties of MoGr.

Using the calculated mass and number fractions we may now use them to approximate the composite material in MERLIN by treating it as a weighted mixture of the constituent elements. This is not the final approach used, however it is available to compare directly with SixTrack. The final approach uses the composite bulk properties for MCS and ionisation, but for point like scattering the number fractions are used to select a weighted random constituent element from the composite. Thus point like scattering is performed using the advanced scattering methods developed in [49] for elements for which the properties are well defined.

For the case of MoGr we treat it as a mixture of pure Mo and C. The mean atomic number $\bar{Z}$ of the composite may be calculated using [45]:

$$\bar{Z} = \sum_i n_i \cdot Z_i, \tag{C.8}$$

where n_i is the number fraction of constituent i, which has atomic number Z_i. The mean atomic mass $\bar{A}$ is similar,

$$\bar{A} = \sum_i n_i \cdot A_i. \tag{C.9}$$

Using these definitions we can define the electron density n_e, mean excitation energy $\bar{I}$, and the radiation length χ_0 for the compound material. The electron density of a compound material is given by:

$$n_e = \frac{\bar{Z}}{\bar{A}} N_a \rho, \tag{C.10}$$

where ρ is the density of the compound, which must be entered by the user as it cannot be calculated. The value of ρ is given by experimental measurements (see [47] for measured novel collimator material properties). The radiation length of a compound material is:

$$\frac{1}{\chi_0} = \sum_i \frac{m_i}{\chi_i}. \tag{C.11}$$

The mean excitation energy is given by equation C.12 [49]:

$$\bar{I} = \sum_i m_i \frac{Z_i}{A_i} \cdot ln \left(1.13 \cdot e^{\left(\frac{I_i}{\frac{\bar{Z}}{\bar{A}}} \right)} \right), \tag{C.12}$$

where the factor of 1.13 is due to stronger atomic bonding in composites [108].

For copper-carbon-diamond (CuCD or copper diamond), we are given the constituents in volume fractions. In this case we may use equation C.14 to calculate the mass fractions:

$$m_i = V_i \frac{\rho_i}{\rho}, \tag{C.13}$$

where V_i is the volume fraction, ρ_i is the density of the constituent, and ρ is the density of the composite. From this the mass fractions shown in Table C.4 are obtained. Both mass fraction and number fraction must satisfy the conditions:

$$m_{tot} = \sum_i m_i = 1, \qquad n_{tot} = \sum_i n_i = 1. \tag{C.14}$$

As these condition are not met by the calculated mass fractions we must normalise the calculated mass fractions to calculate the correct number fractions and other weighted properties.

We compare the properties of these composite materials in MERLIN with those in SixTrack in Table C.5. We note that the differences between MERLIN and SixTrack are the mean free path λ_{tot}, the nuclear slope b_n, and the radiation length χ_0. This is due to differences in the calculation of these quantities between the codes. For the radiation length MERLIN and SixTrack perform a weighted average using equation C.15, from which it is clear that a small difference in the constituent material radiation length or mass fraction will result in a difference in the averaged radiation length.

$$\frac{1}{\chi_0} = \sum_i \frac{m_i}{\chi_i}. \tag{C.15}$$

The difference in the case of MoGr may be explained as in SixTrack this composite is treated as a compound of Mo_2C and graphite, whereas in MERLIN we treat it as a compound of pure carbon and molybdenum. The calculation of the nuclear slope is nearly identical in both codes, and is shown in equation C.16.

$$b_{n\ MERLIN} = 14.1A^{\frac{2}{3}}, \qquad b_{n\ SixTrack} = 14.1A^{0.65}. \tag{C.16}$$

The difference in these expressions explains that of the b_n values in both codes.

In SixTrack the total nuclear reference cross section $\sigma_{pN\ tot}$ is calculated from the mean free path λ_{tot} in the material, which is obtained from the Particle Data Group (PDG) [87], in units of $[\frac{g}{cm^2}]$. Equation C.17 is then used:

V_{Cu}	0.39
V_{CD}	0.60
V_B	0.01
A_{Cu}	63.546
A_{CD}	12.01
A_B	10.811
A_{CuCD}	24.1607
Z_{CuCD}	11.4181
Calculated	
m_{Cu}	0.6471
m_{CD}	0.3489
m_B	0.0044
Normalised	
m_{Cu}	0.6467
m_{CD}	0.3489
m_B	0.0044
n_{Cu}	0.2568
n_{CD}	0.733
n_B	0.0103
ρ_{Cu}	8.96 $\frac{g}{cm^3}$
ρ_{CD}	3.52 $\frac{g}{cm^3}$
ρ_B	2.37 $\frac{g}{cm^3}$
ρ_{CuCD}	5.4 $\frac{g}{cm^3}$

Table C.4: Relative properties of CuCD.

$$\sigma_{pN\ tot} = \frac{A}{N_a \cdot \lambda_{tot}} \cdot 10^{24}, \tag{C.17}$$

where N_a is Avogadro's constant, and the factor of 10^{24} is used to convert from units of cm^2 to barn. In MERLIN (as discussed in Chapter 4) the mean free path λ_{tot} is given from the total cross section, which is a sum of the cross sections of all `ScatteringProcess`es assigned to the collimation process. For a composite in SixTrack, the λ_{tot} value is not available from the PDG (who obtain data from experiments), instead it is taken as a weighted average of the constituent element mean free paths λ_i using the mass fractions as shown in equation C.18:

$$\lambda = \frac{1}{\sum_i \frac{m_i}{\lambda_i}}. \tag{C.18}$$

Then, as for a pure element, $\sigma_{pN\ tot}$ is calculated using equation C.17. MERLIN

Property	MoGr	CuCD	Glidcop	Inermet180
Z				
MERLIN	6.611	11.896	28.824	67.66
SixTrack	6.653	11.898	28.823	67.657
A				
MERLIN	13.44	25.23	63.145	166.7
SixTrack	13.532	25.238	63.149	166.68
$\rho\ [\frac{g}{cm^3}]$				
MERLIN	2.5	5.4	8.93	18
SixTrack	2.5	5.4	8.93	18
$\sigma_{el}\ [\frac{MS}{m}]$				
MERLIN	1	12.6	53.8	8.6
SixTrack	1	12.6	53.8	8.6
$\chi_0\ [m]$				
MERLIN	0.1214	0.03164	0.01443	0.00385
SixTrack	0.11931	0.03162	0.01442	0.00385
$\frac{dE}{dx}$				
MERLIN	0.7188	1.981	2.685	5.576
$b_n\ [\frac{GeV}{c^2}]$				
MERLIN	79.69	121.3	223.59	427
SixTrack	76.665	114.961	208.669	392.137
$\lambda_{tot}\ [m]$				
MERLIN	0.2301	0.1259	0.0892	0.0577
SixTrack	0.2484	0.1356	0.0942	0.0603
MERLIN Modified	0.2257	0.1259	0.0892	0.0577
MERLIN Modified (SixTrack Scattering)	0.2297	0.1283	0.0909	0.0588
MERLIN (SixTrack Scattering)	0.2342	0.1282	0.0909	0.0588

Table C.5: Properties of composite materials in MERLIN and SixTrack [45], after updating cross sections for molybdenum and copper in MERLIN.

takes a similar approach but works from the cross sections, equation C.19 is used to find the reference nuclear cross section (total, inelastic, elastic, or Rutherford):

$$\sigma_{pN} = \sum_i n_i \sigma_{pN\ i}. \tag{C.19}$$

We compare the cross sections in Table C.6. For comparison the values used in MERLIN are overwritten with those used in SixTrack. The resulting mean free paths and cross sections are shown in Table C.5 and Table C.6 respectively, as **MERLIN modified**. We find that the cross sections calculated in MERLIN and SixTrack to be identical for the specified composites, despite the difference in approach.

We note that the proton nucleon cross sections for elastic and single diffractive

Property	MoGr	CuCD	Glidcop	Inermet180
$\sigma_{pN\ tot\ ref}$				
MERLIN	0.3545	0.572	1.246	2.546
SixTrack	0.362	0.572	1.246	2.548
$\sigma_{pN\ I\ ref}$				
MERLIN	0.244	0.370	0.765	1.525
SixTrack	0.247	0.370	0.765	1.473
$\sigma_{pN\ R\ ref}$				
MERLIN	$11.9{\cdot}10^{-5}$	0.000449	0.00151	0.006807
SixTrack	$9.4{\cdot}10^{-5}$	0.000279	0.001385	0.005737
$\sigma_{pN\ tot}$				
MERLIN	0.3827	0.6068	1.293	2.611
MERLIN modified	0.3902	0.6068	1.293	2.613
$\sigma_{pN\ I}$				
MERLIN	0.244	0.370	0.765	1.525
MERLIN modified	0.247	0.370	0.765	1.473
$\sigma_{pN\ E}$				
MERLIN	0.07459	0.1584	0.4217	0.939
MERLIN modified	0.0796	0.1583	0.4217	0.993
$\sigma_{pn\ E}$				
MERLIN	0.04487	0.05536	0.07516	0.1039
MERLIN modified	0.04487	0.05536	0.07516	0.1039
$\sigma_{pn\ SD}$				
MERLIN	0.001879	0.02318	0.03148	0.0435
MERLIN modified	0.001879	0.02318	0.03148	0.0435
σ_R				
MERLIN	$11.9{\cdot}10^{-5}$	0.000449	0.00151	0.006807
MERLIN modified	$9.4{\cdot}10^{-5}$	0.000279	0.001385	0.005737

Table C.6: Cross sections of composite materials in MERLIN and SixTrack citequarantaprivate, after updating cross sections for molybdenum and copper in MERLIN, all values are given in barns.

scattering are not dependent on the reference nuclear cross sections, and so are not affected by the differences between MERLIN and SixTrack.

When using MERLIN scattering with the proper treatment of composite materials, the only effect of the composite properties is seen in the multiple Coulomb scattering and ionisation calculations (`ScatteringModel::EnergyLoss()` and `ScatteringModel::Straggle()`. These functions rely on the electron density, plasma density, mean excitation energy, and radiation length, which are all treated with standard methods for composites. The mean free path is also used as a step length in the collimation process

`CollimateProtonProcess::DoScatter()`. We have shown that the radiation length and mean free path are comparable with those in SixTrack. The difference in radiation length is small, and is due to the difference in stored radiation lengths for constituent elements. The difference in mean free paths is more noticeable, particularly for MoGr. As MERLIN uses its own advanced scattering routines to calculate the cross sections (despite using similar reference cross sections as SixTrack), it is inevitable that this difference will occur, and is in fact understood to be an improvement on SixTrack.

Appendix D

MERLIN 5-01

In this appendix brief code snippets will be used to help describe and explain MERLIN, the colour convention in these snippets is described in Fig. D.1.

```
1   //Comments are shown in dark green
    //Units are shown in lime, and C++ keywords in blue
3   Unit;
    double length = 26.67 * km;
5
    //Base classes are shown in azure, and user classes in orange
7   Base_Class User_Class;
    MADInterface* myMADInterface = new MADInterface();
9
    //Functions are shown in pink, strings in gray
11  Function();
    string test = "string";
13  myMADInterface->TreatTypeAsDrift("DRIFT");
```

Figure D.1: Colour convention used for code snippets.

To define a beam energy of 7 TeV for example, one would use the code shown in Fig D.2.

The following sections describe how a collimation simulation may be written using

```
double beam_energy = 7 * TeV;
```

Figure D.2: Example of the use of units in MERLIN.

MERLIN, indicating differences between the LHC and HL-LHC requirements, and new modifications. A MERLIN simulation consists of three main parts: creating an accelerator model, defining a beam, and selecting and using a tracker to transport the defined beam through the created accelerator. Physics processes may also defined and assigned to the tracker.

D.1 Accelerator Construction

MERLIN stores an accelerator as an `AcceleratorModel` object, this requires three input files for the purpose of LHC and HL-LHC collimation, the first for an accurate model of the individual components of the accelerator, the second to define the apertures of the machine, and the third to define collimator parameters. In order to translate these inputs into a MERLIN `AcceleratorModel`, a number of input interfaces and configuration utilities are required.

D.1.1 Accelerator Model

The model must define the type and location of accelerator elements, which MERLIN stores as `AcceleratorComponent`s. The `AcceleratorModel` is an ordered vector of `AcceleratorComponent`s, which is used by the `ParticleTracker` to set the integrators that describe the paths taken by individual particles as they travel through the elements.

D.1.1.1 AcceleratorComponent

Each `AcceleratorComponent` contains an EM field, a geometry, an aperture, and a wake potential object (the latter of which will not be used in this thesis). Special cases exist, for example a `Collimator` also contains a jaw material which is required for scattering. Figure D.3 shows the inheritance of the standard `AcceleratorComponent`s, both `Collimator` and `HollowElectronLens` elements are tracked as, and therefore derived from, the `Drift` component (a vacuum pipe with no field).

D.1.1.2 MADInterface

A standard MADX [85] thick lens TFS table is passed through the `MADInterface` class to extract the `AcceleratorComponent`s. `MADInterface` strips the header then reads the column headers to see what is defined in the TFS table, before iterating through

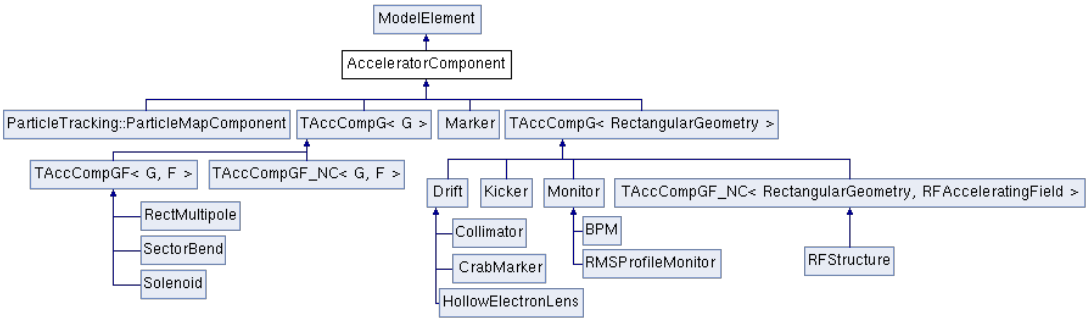

Figure D.3: Accelerator components currently available in MERLIN.

each element and creating the appropriate `AcceleratorComponent`, and appending it to the `AcceleratorModelConstructor`, a standalone class used in `MADInterface`.

A bug exists in the creation of thick lens RFCAVITY elements in MADX, such that the length of the cavity is incorrect. MADX creates an RFCAVITY and DRIFT pair, such that the total length is correct. However the RFCAVITY length is not a multiple of $\frac{\lambda}{2}$, where λ is the wavelength of the cavity wave. This is a requirement of the cavity, as well as the corresponding integrator, if this condition is not met then MERLIN will throw an exception. In order to correct this a flag has been added to calculate the correct number of cavity cells, each with length $\frac{\lambda}{2}$, this can be set using the `SetSingleCellRF()` function. Any left over length is appended as a DRIFT.

The TFS table may contain the element apertures, however in the LHC and HL-LHC case the apertures do not correspond exactly to the elements, thus a separate input file is required. Construction of apertures using `MADInterface` may be enabled or disabled using the `ConstructApertures()` function. Finally the `AcceleratorModel`, a vector of ordered `AcceleratorComponents`, may be extracted from `MADInterface` using the `ConstructModel()` function, which returns a pointer to the created `AcceleratorModel`. The code required to construct an `AcceleratorModel` using `MADInterface` is shown in Fig. D.4.

MERLIN contains a number of standard components, detailed in Table D.1. Any element may be treated as a drift using the `MADInterface::TreatTypeAsDrift()` function, though this is unwise for certain magnets. MERLIN currently handles a small number of elements as drifts as standard because there is no integrator to perform the expected function, or the expected function cannot be performed using an integrator (for example the hollow electron lens), these elements are listed in Table D.2.

```
MADInterface myMADInterface = new MADInterface ("LatticeFile.tfs");
  myMADInterface->TreatTypeAsDrift{"OCTUPOLE"};
  myMADInterface->ConstructApertures(false);
  myMADInterface->SetSingleCellRF(true);

AcceleratorModel* myAcceleratorModel = myMADInterface->ConstructModel();
```

Figure D.4: Construction of an `AcceleratorModel` using a MADX TFS table input file and the `MADInterface` class. In this example we treat octupole elements as drifts, do not construct apertures from the 'LatticeFile.tfs', and use the single cell RF flag to fix the MADX thick lens RFCAVITY length bug.

D.1.2 Apertures

For collimation, all `AcceleratorComponents` must have apertures, these may not necessarily correspond 1:1 with accelerator components, this is the case in the LHC and thus a separate MADX generated aperture list is required. In the case of the HL-LHC this file is constructed by hand and stored in a similar MADX format. MERLIN contains a number of aperture types, as shown in Figure D.5.

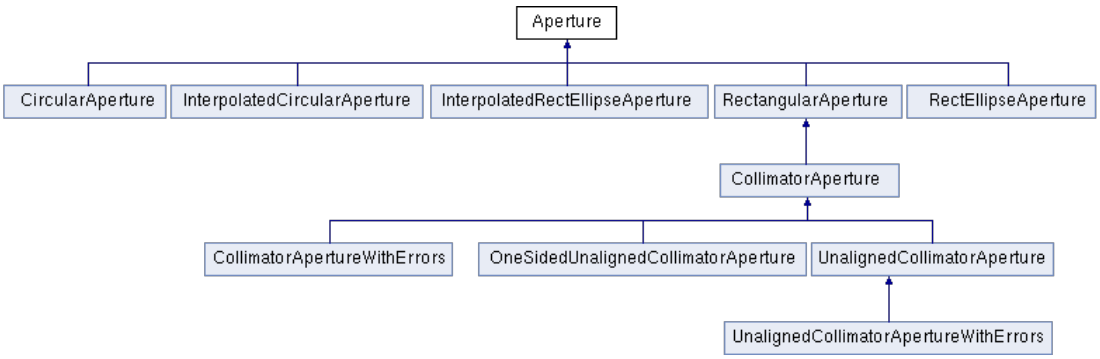

Figure D.5: Aperture types and inheritance.

All apertures contain a `PointInside()` function, which takes the spatial co-ordinates of a particle (x, y, z), and checks if those co-ordinates are inside the aperture. This function is used by the collimation process to select which particles should undergo scattering, or be lost and thus removed from the bunch after recording the loss location. The RectEllipse type of aperture is illustrated in Fig D.6.

Component	MADX name	MERLIN name
Vacuum pipe	DRIFT	Drift
RF Cavity	LCAV	SWRFStructure
RF Cavity	RFCAVITY	SWRFStructure
Collimator	RCOLLIMATOR	Collimator
Collimator	ECOLLIMATOR	Collimator
Collimator	COLLIMATOR	Collimator
Rectangular Dipole	RBEND	SectorBend[1]
Sector Dipole	SBEND	SectorBend[1]
Multipole	MULTIPOLE	-[2]
Vertical Corrector	YCOR	YCor
Vertical Kicker	VKICKER	YCor
Horizontal Corrector	XCOR	XCor
Horizontal Kicker	HKICKER	XCor
Quadrupole	QUADRUPOLE	Quadrupole
Skew Quadrupole	SKEWQUAD	SkewQuadrupole
Solenoid	SOLENOID	Solenoid
Hollow Electron Lens	-	HollowElectronLens
Sextupole	SEXTUPOLE	Sextupole
Octupole	OCTUPOLE	Octupole
Skew Sextupole	SKEWSEXT	SkewSextupole
Crab Cavity	CRABRF	TransverseRFStructure[3]
Monitor	MONITOR	BPM, RMSProfileMonitor[4]
Marker	MARKER	Marker

Table D.1: Accelerator components and their MADX TFS and MERLIN names. [1] With appropriate pole face rotation pre and post element. [2] MERLIN uses the lowest order non-zero magnet coefficient to set a MADX MULTIPOLE to the lowest order magnet possible. [3] Crab Cavities are not yet supported in MERLIN, though an integrator has been constructed and will be implemented in the near future. [4] The `MADInterface` reads the suffix of 'MONITOR', if this is 'BPM' a beam position monitor (BPM) is created, if the suffix is 'WS' an RMSProfileMonitor is created, an unknown monitor type defaults to a BPM.

Component	MADX name	MERLIN name
Vacuum pipe	DRIFT	Drift
Generic Kicker	KICKER	Drift
Generic Kicker	TKICKER	Drift
Instrumentation Device	INSTRUMENT	Drift
Placeholder	PLACEHOLDER	Drift

Table D.2: Accelerator components treated as drifts by default in MERLIN, with their corresponding MADX TFS names.

D.1.3 Aperture Configuration

The `ApertureConfiguration` class was added to the loss map version of MERLIN, and modified for the HEL and merged codes. The input TFS table contains columns

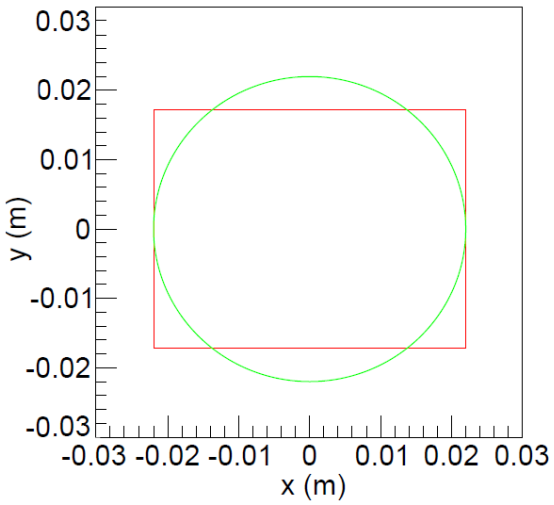

Figure D.6: RecTellipse aperture in real space, showing the elliptical component in green, and rectangular component in red. Adapted from [49].

as depicted in Table D.3.

KEYWORD	NAME	PARENT	S	L	APER_1	APER_2	APER_3	APER_4	APER_TYPE

Table D.3: Aperture TFS table column headers.

When merging `ApertureConfiguration` it was updated to take account of the HL-LHC aperture files, which assume that all elements have a RectEllipse aperture - in which case the final column of the file is omitted. This may be activated using the `SetAllRectEllipse()` function. After giving the `ApertureConfiguration` class the input file, the `ConfigureApertures()` function takes a pointer to the `Accelerator Model`, and sets the apertures for all elements except `Collimators` using this input file, as shown in Fig. D.7.

The `ApertureConfiguration` class can create interpolated apertures. This is required for two reasons, firstly the apertures of the machine do not necessarily correspond 1:1 with the elements of the machine, thus more than one apertures may be defined for a single element, and secondly because the aperture file gives the size of apertures at discrete positions in the machine, rather than a continuous model. The workflow of the `ConfigureElementApertures` function is shown in Fig. D.8.

All collimator apertures are set using the `CollimatorDatabase` class which is discussed in a later subsection.

```
ApertureConfiguration* myApertureConfiguration = new
    ApertureConfiguration("ApertureInput.TFS");

myApertureConfiguration->SetAllRectEllipse();
myApertureConfiguration->ConfigureElementApertures(myAcceleratorModel)
    ;
```

Figure D.7: Construction of apertures using a MADX TFS table aperture input file and the `ApertureConfiguration` class. In this example we use the `SetAllRectEllipse` flag to use an HL-LHC aperture file.

Element Name	S	|horizontal positive|	|horizontal negative|	|vertical positive|	|vertical negative|

Table D.4: ApertureSurvey column headers.

D.1.3.1 ApertureSurvey

The `ApertureSurvey` class allows the user to output the complete aperture of the machine being simulated in MERLIN, in the horizontal and vertical planes. The user may specify either a number of steps to sample per element, or an increment length. The `ApertureSurvey` object must simply be created, it then automatically generates an output file (for which the format is shown in Table D.4) in the directory given as an argument to the constructor. The `ApertureSurvey` constructor and corresponding arguments are shown in Fig. D.9. If the number of points per element is greater than 0, this is used to survey the machine aperture, otherwise the specified step size (in metres) is used. Without these values the default method is to survey the aperture of the machine at 10 cm steps per element.

This output allows the user to directly compare the MERLIN aperture with that of the input file, or another code. The aperture for the start of the 6.5 TeV LHC lattice is compared with the MADX aperture file in Fig. D.10. We observe small differences due to the fact that apertures are continuous in MERLIN, whilst the MADX input file only gives the aperture at discrete points (though it is plotted using a linear interpolation between these points).

D.1.4 Lattice Functions

The `LatticeFunctionTable` class takes the `AcceleratorModel` and beam energy to calculate the lattice functions by particle tracking. `LatticeFunctions` are added

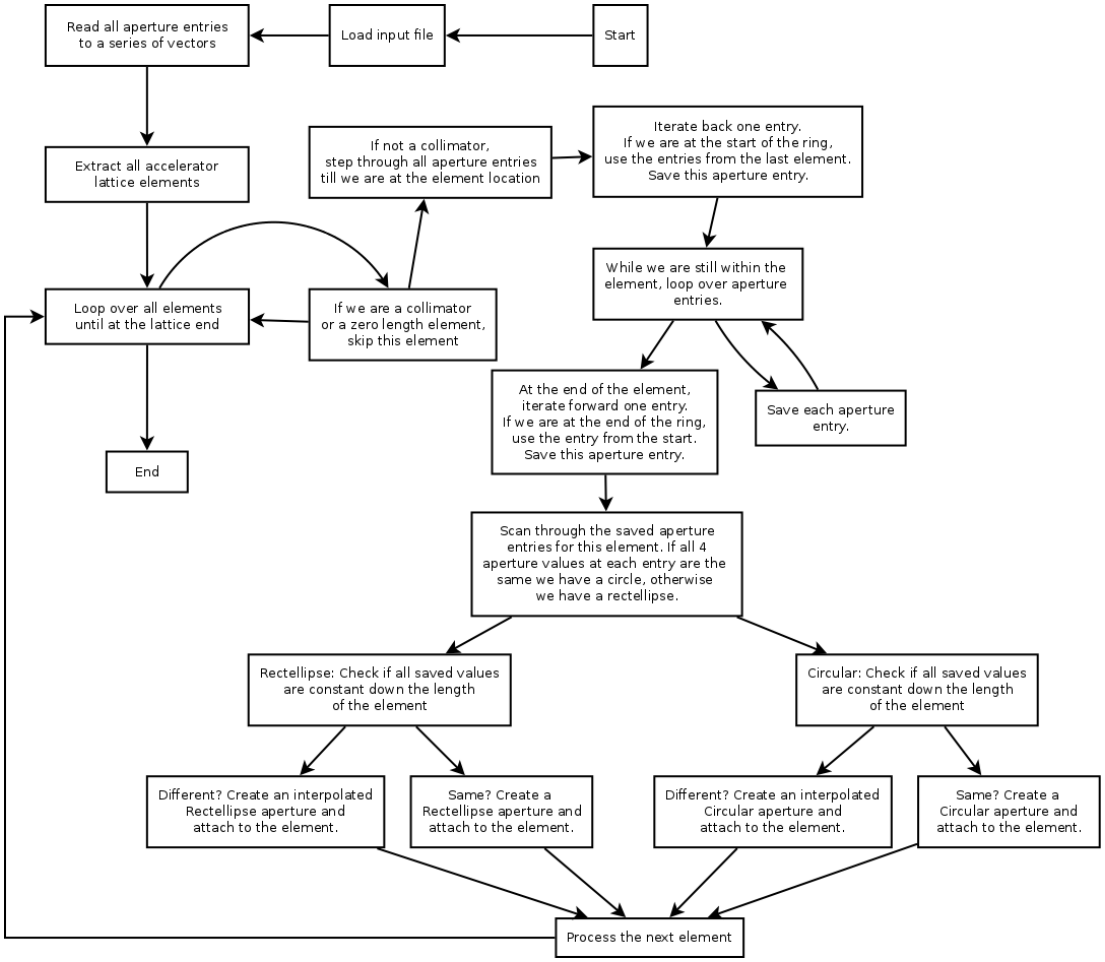

Figure D.8: Workflow of the `ApertureConfiguration::ConfigureElementApertures()` function. This does not take into account the updated `SetAllRectEllipse` flag. From [49]

```
1    ApertureSurvey(AcceleratorModel* model, string directory, double
         step_size=0.1, size_t points_per_element=0);
```

Figure D.9: The `ApertureSurvey` constructor, currently no other function is required to use this class.

to this table either individually, or using the `LatticeFunctionTable::UseDefault Functions()` function which is called in the constructor. Each lattice function has 3 indices i, j, k, the combinations and their corresponding parameters are detailed in

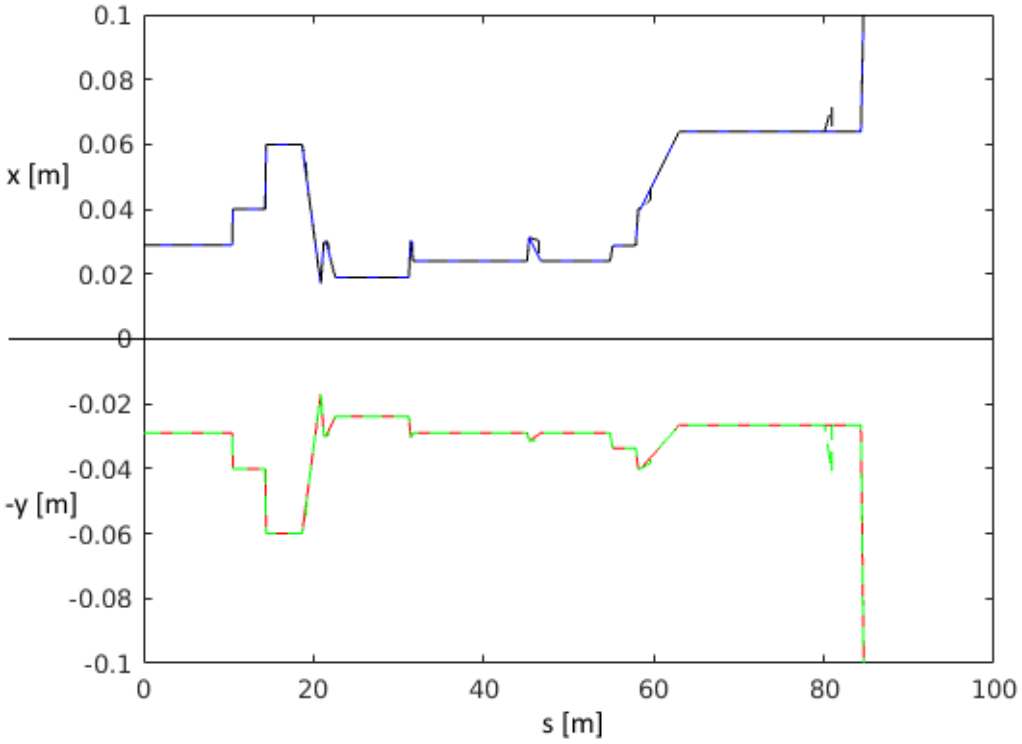

Figure D.10: Comparison of MADX apertures (vertical = red, horizontal = blue) with MERLIN apertures (vertical = green dashed, horizontal = black dashed) for the first 100 m of the LHC using 6.5 TeV optics. The positive horizontal aperture and inverted positive vertical aperture are shown.

Table D.5.

The lattice functions are calculated from eigenvectors of the one turn map, which is not complete without the action of RF cavities, which may not always be used. In order to compensate for this the `ScaleBendPath()` function is used to adjust the ct and δ values such that the lattice functions may be calculated with no convergence errors. For collimation simulations the MADX TFS table used to create the `AcceleratorModel` usually contains the lattice functions, which are also calculated using a 4D method.

The functions of the lattice shown in Table D.5 are stored in the `LatticeFunction Table`, where they are accessed using the `LatticeFunctionTable::Value()` function, which takes the indices i, j, k and a fourth index which corresponds to the position of the element in the `AcceleratorModel` - this is the number of the element in the vector, not its physical position. An example of the use of this class is shown in Fig. D.18,

i	j	k	**Parameter**
1	0	0	Closed orbit x
2	0	0	Closed orbit x'
3	0	0	Closed orbit y
4	0	0	Closed orbit y'
5	0	0	Closed orbit ct
6	0	0	Closed orbit δ
1	1	1	β_x
3	3	2	β_y
1	2	1	$-\alpha_x$
3	4	2	$-\alpha_y$
0	0	1	Fractional phase advance μ_x
0	0	2	Fractional phase advance μ_y

Table D.5: `LatticeFunctionTable` indices and corresponding parameters.

where the β_x calculated in MERLIN is plotted against that calculated in MADX in Fig. D.11.

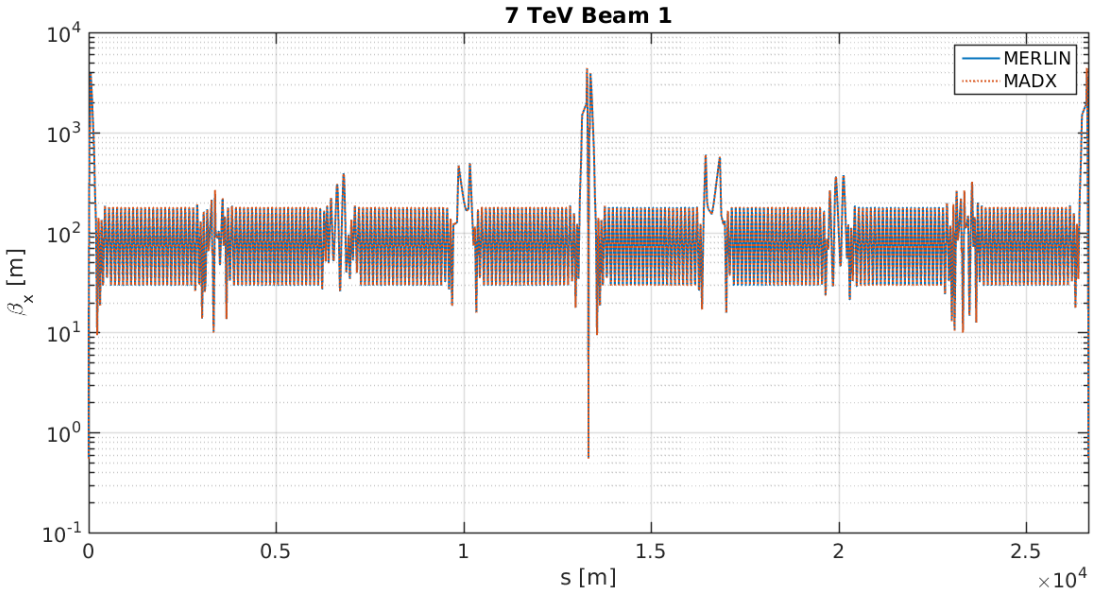

Figure D.11: β_x calculated using the `LatticeFunctionTable` class in MERLIN (blue) compared with that calculated in MADX (orange), for beam 1 of the nominal LHC.

The `Dispersion` class is used to calculate the dispersion of a given lattice, although this functionality is given by the `LatticeFunctionTable`, a more accurate method is

used in this class. The use of this class is shown in Fig. D.18, and an example of the output is given in Fig. D.12.

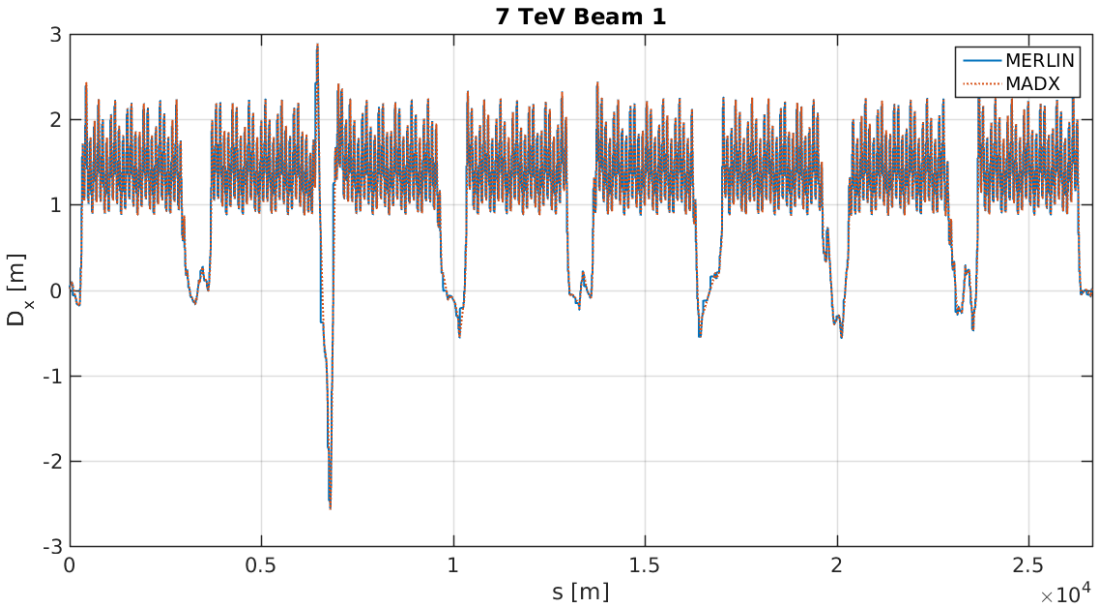

Figure D.12: The dispersion in the x plane D_x calculated using the `Dispersion` class in MERLIN (blue) compared with that calculated in MADX (orange), for beam 1 of the nominal LHC.

D.1.5 Phase Advance

The `PhaseAdvance` class has been created to calculate and return the phase advance in x or y at a given element, or between any two elements in the lattice. This can be calculated using either the name or the number of the element in the `AcceleratorModel`. The `PhaseAdvance` class takes the `LatticeFunctionTable` and `AcceleratorModel` as arguments, and iterates through the lattice in order to sum the fractional phase advance from the `LatticeFunctionTable`. This class is required for the crab cavity failure process which is experimental and outside the scope of this thesis, however the functionality may be helpful for other processes.

D.1.5.1 Collimator Database

The `CollimatorDatabase` class reads a separate file which defines the collimator settings.

Name	Jaw Half Gap $[\sigma]$	Perpendicular Plane ($\perp$) Half Gap $[\sigma]$	Jaw Rotation α $[rad]$	Material Symbol

Table D.6: Column headings for the collimator setup file read by `CollimatorDatabase` when `use_sigma` is true.

Name	Half Gap $[\sigma]$	$\perp$ Half Gap $[\sigma]$	α $[rad]$	Tilt$_1$ $[rad]$	Tilt$_2$ $[rad]$	Material

Table D.7: Column headings for the collimator setup file read by `CollimatorDatabase` when `use_sigma` is false.

`CollimatorDatabase` takes the collimator file name, a pointer to `MaterialDatabase` (see later section), and a boolean `use_sigma` that tells the class the format of the input file. If true the column headings shown in Table D.6 are expected, if false the column headings shown in Table D.7 are expected, adding any tilting of the collimator jaws.

`CollimatorDatabase` extracts all collimators in the `AcceleratorModel` and iterates through them. The corresponding lattice functions are obtained from the `LatticeFunctionTable`, and one of the above methods is used to create a collimator aperture and attach it to the `Collimator` element. The `Collimator` also contains a `Material` which is set here using the `MaterialDatabase`.

The majority of collimators consist of two movable jaws, however some LHC collimators, for example the dump protection TCDQs are one-sided. The names of these collimators are stored in the `CollimatorDatabase`, and if one of these collimators is selected, a single sided aperture is created in the appropriate plane. In the near future this class will be updated for FCC (Future Circular Collider) [109] apertures [110], and will also include a function to input one-sided collimator names. If a collimator that exists in the lattice is not defined in the collimator input file, it will have no aperture.

Figure D.13 shows an example of how to use the `CollimatorDatabase`.

```
1  CollimatorDatabase* myCollimatorDatabase = new CollimatorDatabase( "
       collimator_file.dat", myMaterialDatabase, true);

3  myCollimatorDatabase->MatchBeamEnvelope(true);
   myCollimatorDatabase->UseMidJawHalfGap();
```

Figure D.13: `CollimatorDatabase` usage with the `UseMidJawHalfGap` function in order to set a constant collimator aperture for the length of each collimator, with half gaps calculated using the beta functions at the centre of the collimator.

ID	Name	α	β_x	β_y	g	Material	L	σ_x	σ_y	θ_1	θ_2	$n\sigma$

Table D.8: FLUKADatabase output file column headings.

D.1.5.2 CollimatorSurvey

`CollimatorSurvey` is similar to the `ApertureSurvey` class, however only collimator apertures are surveyed and output. This allows a detailed survey (for example at increments of mm) of the collimator jaw openings. This provides a mechanism to show the difference between collimator jaw setting methods, the `MatchBeamEnvelope` and `MidJaw` options are compared using the `CollimatorSurvey` for the primary collimator in IR7 in Fig. 4.2.

D.1.5.3 FLUKA Database

The `FLUKADatabase` is an output designed for future use of MERLIN with FLUKA, though not currently pursued, it is useful for certain direct comparisons with SixTrack. This file includes columns listed in Table D.8, for all non-zero length collimators. α is the collimator rotation angle, the ID is an arbitrary integer that is stored in the `Collimator` and used for outputting `FlukaLosses`, L is the collimator length, σ is the value of 1 beam σ at the collimator, g is the half gap in the primary plane of the collimator, $n\sigma$ is the jaw opening in units of beam σ, and θ is the jaw tilt. All values are given either at the start of the collimator, or the middle, depending on whether or not the `UseMidJawHalfGap()` option has been used.

As all required values are calculated in, or pass through the `CollimatorDatabase` class, this is where the `FLUKADatabase` data is stored and output using the function `CollimatorDatabase::OutputFlukaDatabase(std::ostream* os)`.

D.2 Materials

The `Material` constructor is shown in Fig. D.14, it is important to remember that when using MERLIN scattering a number of properties not included in the constructor are required; the total nuclear cross section, the nuclear slope, and the mean excitation energy. Some of these attributes may be calculated using related functions that form part of the class, these are shown below the constructor in Fig. D.14.

All `Material` properties are private, they may be set using the constructor, or the corresponding setting function. The setting function for a property is `Set` followed

```
    Material(string name0, string sym0, double A0, int AtomicNumber0,
        double Sigma_E0, double Sigma_I0, double Sigma_R0, double dEdx0,
        double X00, double Density0, double Conductivity0);

    double CalculateSixtrackTotalNucleusCrossSection();
    double CalculateSixtrackNuclearSlope();
    double CalculateMeanExcitationEnergy();
```

Figure D.14: The `Material` constructor, taking the name, symbol, atomic mass, atomic number, elasic cross section, inelastic cross section, Rutherford cross section, stopping power, radiation length, density, and conductivity as arguments. Functions to calculate material attributes required for MERLIN scattering.

```

Material* Be = new Material("Beryllium", "Be", 9.012182, 4, 0.069,
    0.199, 0.000035, 0.55, 651900, 1848, 3.08E7);
Be->SetSixtrackTotalNucleusCrossSection(0.268);
Be->SetSixtrackNuclearSlope(74.7);
Be->SetMeanExcitationEnergy(63.7*eV);
Be->SetElectronDensity(Be->CalculateElectronDensity());
Be->SetPlasmaEnergy(Be->CalculatePlasmaEnergy());

Material* Pb = new Material();
Pb->SetAtomicNumber(82);
Pb->SetAtomicMass(207.2);
Pb->SetName("Lead");
Pb->SetSymbol("Pb");
Pb->SetSixtrackTotalNucleusCrossSection(2.960);
Pb->SetSixtrackInelasticNucleusCrossSection(1.77);
Pb->SetSixtrackRutherfordCrossSection(0.00907);
Pb->SetSixtrackdEdx(3.40);
Pb->SetConductivity(4.8077E6);
Pb->SetRadiationLength(Pb->CalculateRadiationLength());
Pb->SetDensity(11350);
Pb->SetSixtrackNuclearSlope(455.3);
Pb->SetMeanExcitationEnergy(823.0*eV);
Pb->SetElectronDensity(Pb->CalculateElectronDensity());
Pb->SetPlasmaEnergy(Pb->CalculatePlasmaEnergy());
```

Figure D.15: An example of constructing Beryllium using the `Material` constructor, setting functions, and calculation functions. An example of constructing Lead using the `Material` setting functions and calculation functions.

by the MERLIN name detailed in Table 4.2, for example `SetDensity()`, a number of these functions are shown in Fig. D.15. The access function is similar but has the prefix `Get`, for example `GetDensity()`. A number of functions exist to calculate the following properties: electron density, plasma energy, mean excitation energy, nuclear slope, radiation length, total nuclear cross section, and the inelastic nuclear cross section.

The `MaterialDatabase` class provides a dictionary of standard materials (listed in Table 4.3) that may be used without the user defining the corresponding `Material` object. The `FindMaterial()` function allows the user to search the dictionary by the material symbol, and returns a pointer to the `Material` object. This is shown in Fig. D.16, where the symbol for copper is used to return a pointer to the corresponding `Material`. The `VerifyMaterial()` function checks that all required properties are set and non-zero, it returns a `true` boolean if this is the case, or a `false` boolean if not, as well as outputting the discrepancy.

```
1
    MaterialDatabase* myMaterialDatabase = new MaterialDatabase();
3   Material* myMaterial = myMaterialDatabase->FindMaterial("Cu");
    bool verified = myMaterial->VerifyMaterial();
```

Figure D.16: An example of using the `MaterialDatabase` to find and return a pointer to the `Material` object for copper.

The construction of the `CompositeMaterial` (detailed in Chapter 4) copper-carbon-diamond (CuCD) in the `MaterialDatabase` dictionary is shown in Fig. D.17.

D.2.1 `BeamData`

`BeamData` provides a data structure for definition of the 6D beam phase space. Using the `LatticeFunctionTable` the user may define parameters at the injection position, which are fed to the `ParticleBunchConstructor`. The components of `BeamData` are shown in Table D.9. To improve user control over the initial bunch, a number of parameters were added to the `BeamData` class to set minimum and maximum values for the initial co-ordinates.

The use of the `LatticeFunctionTable` to set `BeamData` components is shown in Fig. D.18, we note that in order to define a matched initial bunch, components labelled in Table D.9 must be set. A number of these parameters are set to zero by default.

Using the `ParticleBunchConstructor` as shown in Fig. D.20 is preferable as the

```
CompositeMaterial* CuCD = new CompositeMaterial();
  CuCD->SetName("CopperCarbonDiamond");
  CuCD->SetSymbol("CuCD");

  CuCD->AddMaterialByMassFraction(CD,0.3489);
  CuCD->AddMaterialByMassFraction(Cu,0.6467);
  CuCD->AddMaterialByMassFraction(B,0.0044);

  CuCD->SetDensity(5400);
  CuCD->SetConductivity(12.6E6);
  CuCD->Assemble();
  CuCD->VerifyMaterial();
```

Figure D.17: The construction of the `CompositeMaterial` CuCD in the `Material Database` dictionary.

```
  LatticeFunctionTable* myTwiss = new LatticeFunctionTable(myAccModel,
      beam_energy);
  myTwiss->ScaleBendPathLength(1e-22);
  myTwiss->Calculate();

  Dispersion* myDispersion = new Dispersion(myAccModel, beam_energy);
  myDispersion->FindDispersion(start_element_number);

  BeamData myBeam;
  myBeam.beta_x = myTwiss->Value(1,1,1,start_element_number)*meter;
  myBeam.x0     = myTwiss->Value(1,0,0,start_element_number);
  ...
  myBeam.Dx     = myDispersion->Dx;
  myBeam.Dy     = myDispersion->Dy;
```

Figure D.18: Calculation of lattice functions using the `LatticeFunctionTable` and `Dispersion` classes, and accessing these values to set properties of the initial distribution via the `BeamData` class. `start_element_number` is an integer corresponding to the position of the start element in the `AcceleratorModel`.

bunch is matched to the accelerator lattice.

The `BunchFilter` class may be created and applied to the bunch construction process if the user wishes to perform specific operations on the initial distribution. A child of `BunchFilter`, the `HorizontalHaloParticleBunchFilter` class is used in Fig. D.20 to remove any part of the bunch that occupies $\pm 4\sigma_x$, in order to avoid simulating stable particles for a loss map simulation.

Component(s)	Parameter(s)
x0, xp0, y0, yp0, ct0, dp0	Beam centroid[1]
beta_x, beta_y, alpha_x, alpha_y	Lattice functions[1]
emit_x, emit_y	Emittances[1]
sig_dp	Relative energy spread
sig_z	Bunch length
p0	Reference momentum[1]
c_xy, c_xyp, c_xpy, c_xpyp	x-y coupling
Dx, Dxp, Dy, Dyp	Dispersion[1]
charge	Particle charge[1]
min_sig_x, max_sig_x	Minimum and maximum beam size in x (in σ)[2]
min_sig_y, max_sig_y	Minimum and maximum beam size in y (in σ)[2]
min_sig_z, max_sig_z	Minimum and maximum bunch length[2]
min_sig_dp, max_sig_dp	Minimum and maximum energy deviation[2]

Table D.9: Components of the `BeamData` class. [1] Must be specified. [2] units are dependent on the type of distribution selected in the `ParticleBunchConstructor`.

```
ProtonBunch* myBunch;
ParticleBunchConstructor* myBunchCtor;

// Input file streams for multiple input bunches
ifstream* bunch_input_core = new ifstream("core_bunch_input.txt");
ifstream* bunch_input_halo = new ifstream("halo_bunch_input.txt");

// Select one input bunch
istream* is = bunch_input_core;

myBunch = new ProtonBunch(beam_energy, myBeam.charge, *is);
myBunch->SetMacroParticleCharge(myBeam.charge);
```

Figure D.19: Construction of a `ProtonBunch` using an input file.

D.3 Tracker

The `ParticleTracker` class performs the tracking in MERLIN, using the transport process template function to define a `BunchProcess` which uses `ComponentTrackers` to provide the transfer maps.

A collection of integrator classes, the `ComponentTracker` is used to construct the tracker fed to the `ParticleTracker` for each type of element. These individual integrators perform the mathematical transformation of the particle bunch. MERLIN contains

```
    ProtonBunch* myBunch;
    ParticleBunchConstructor* myBunchCtor = new ParticleBunchConstructor(
        myBeam, n_part, HELHaloDistribution);

    HorizontalHaloParticleBunchFilter* hFilter = new
        HorizontalHaloParticleBunchFilter();
      double h_offset = myTwiss->Value(1,0,0,start_element_number);
      double tcpsig = 266E-6;
      hFilter->SetHorizontalLimit(4*tcpsig);
      hFilter->SetHorizontalOrbit(h_offset);

      myBunchCtor->SetFilter(hFilter);

  myBunch = myBunchCtor->ConstructParticleBunch<ProtonBunch>();
  delete myBunchCtor;
```

Figure D.20: Construction of a `ProtonBunch` using the `ParticleBunchConstructor`, and a `HorizontalHaloParticleBunchFilter`.

a number of integrators, which are can be grouped to select a tracking method.

The tracker takes an `AcceleratorComponent` iterator for the `AcceleratorModel`. This can either be a linear `Beamline` (*e.g.* a linear accelerator), or a `RingIterator` for a repeating lattice (*e.g.* a synchrotron). The tracker also takes the particle bunch, as shown in Fig. D.21.

The `ParticleTracker` contains a `ProcessStepManager` class, which manages all processes attached to the simulation, this includes the particle tracking, as well as any other physics processes that have been attached to it.

D.3.1 Integrator Sets

Integrators define the tracking for a particle through a single element of the accelerator lattice, and can be stored using any method that is compatible with the defined particle bunch co-ordinates. Integrators are divided into a number of sets, and `simple` integrators exist for monitors, markers, solenoids and sector bends.

The `SetIntegratorSet()` function was added by Sam Tygier to enable selection of the desired set of integrators used for a tracker, as shown in Fig. D.21. If no integrator set is selected, the default `TRANSPORT` set is used.

```
AcceleratorModel::RingIterator beamline = myAccModel->GetRing(
    start_element_number);
ParticleTracker* myParticleTracker = new ParticleTracker(beamline,
    myBunch);
bool symplectic = true;
if(symplectic)
myParticleTracker->SetIntegratorSet(new ParticleTracking::SYMPLECTIC::
    StdISet());
else
myParticleTracker->SetIntegratorSet(new ParticleTracking::TRANSPORT::
    StdISet());

myParticleTracker->Track(myBunch);
```

Figure D.21: Defining a `ParticleTracker` with a circular accelerator iterator, or `RingIterator`, and selecting the `SYMPLECTIC` or `TRANSPORT` integrators.

D.3.2 `TRANSPORT`

The `TRANSPORT` integrators are described in Chapter 2. Individual integrators are defined using this method for; drifts, sector bends, rectangular multipoles, quadrupoles, sextupoles, and pole face rotations. The `THIN_LENS` integrators for kickers and SWRF (standing wave radio frequency) cavities are used in the `TRANSPORT` integrator set, as well as the `simple` solenoid integrator.

This set has been used for the majority of past MERLIN simulation studies. In the case of the ILC (a single pass machine), and loss map simulations which require 200 turns of the LHC, the `TRANSPORT` integrator is valid. However for long term simulations, such as those performed for the HEL, it is not recommended because the integration method does not preserve the Hamiltonian of the system.

This integrator set is used when calculating the lattice functions using the `LatticeFunctionTable` class as mentioned previously. An example of the computed closed orbit is shown in Fig. D.22.

D.3.3 SYMPLECTIC

This integrator set includes symplectic integrators for: drifts, thin multipoles, sector bends, quadrupoles, pole face rotations, combined function sector bends, RF structures, and SWRF structures. The `simple` integrators for monitors, markers, solenoids, and travelling wave radio frequency cavities, are included in this set, though the latter is not currently used.

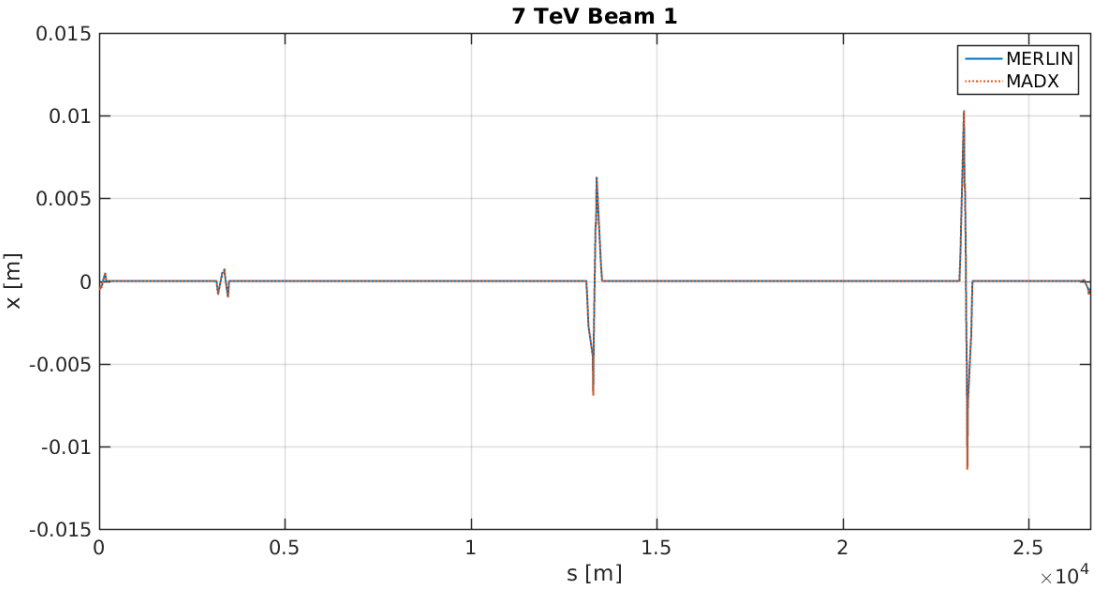

Figure D.22: The closed orbit in x calculated using the `LatticeFunctionTable` class in MERLIN (blue) compared with that calculated in MADX (orange), for beam 1 of the nominal LHC.

D.3.4 Synchrotron Motion

Synchrotron motion occurs when SWRF structures are operational in the machine. It is important to note that these integrators affect the calculation of the lattice functions. When using MADX, the lattice functions are generated in 4 dimensions (x, x', y, y'), thus in order to compare to this and use the same settings (for example collimator jaw half-gaps are proportional to $\beta^{\frac{1}{2}}$) the `klystron` control class may be used. This can be attached to multiple RF cavities, allowing the user to change the RF voltage and phase of those cavities, its use is shown in Fig. D.23.

D.3.5 Tracking Output

The particle tracker can take a `SimulationOutput` class that can be used to output, for example, particle tracks throughout the simulation, or for a selected range. The `TrackingOutputAV` class was created as a child of `SimulationOutput` to output individual particle tracks in order to observe the paths taken by particles that have scattered differently in the collimator jaws. This tool was also used to identify bugs in the `SYMPLECTIC` tracker. Data at the start of every element is stored and output.

The `SetTurn()` function may be used to select a single turn for which particles

```
// After construction of the accelerator model, create a vector to hold
    RF cavities
std::vector<RFStructure*> RFCavities;
// Extract only the ACS type (LHC naming convention)
myAccModel->ExtractTypedElements(RFCavities,"ACS*");
// Attach these cavities to the control class
Klystron* myKlystron = new Klystron("KLY1",RFCavities);
// Set variables
myKlystron->SetVoltage(0.0);
myKlystron->SetPhase(pi/2);
...
// After calling LatticeFunctionTable::Calculate(), set voltage in MV
myKlystron->SetVoltage(2.0);
```

Figure D.23: Use of the `Klystron` control class to set the voltage and phase of a selection of RF cavities.

are tracked, or the `SetTurnRange()` function specifies two turns defining a range in which the data is output. A single turn in this manner begins at the start of the `AcceleratorModel`, and ends at the end of the `AcceleratorModel`, being incremented when $s = 0$. The `SetSRange()` function may be used to limit output to a range in the s co-ordinate of the accelerator. Figure D.24 shows an example of the usage of this class, outputting only for the first turn, in the s range that corresponds to the entire LHC lattice. We note that if we use these commands for a loss map simulation, the tracking output will begin at the injection position in the betatron collimation insertion (s $\approx$ 19000 m, where s = 0 is defined to be at IP1).

```
TrackingOutputAV* myTrackingOutputAV = new TrackingOutputAV("
    TrackingOutput.dat");
 myTrackingOutputAV->SetSRange(0, 27000);
 myTrackingOutputAV->SetTurn(1);
 myTrackingOutputAV->output_all = 1;

 myParticleTracker->SetOutput(myTrackingOutputAV);
```

Figure D.24: Use of the `TrackingOutputAV` class in MERLIN to output particle tracks.

The file name (including the file path if required) is set in the `TrackingOutputAV` constructor, and the output format is shown in Table D.10. Note that the `output_all` component of `SimulationOutput` must be set to 1 for the output to be performed.

Particle ID	Turn	s	x	x'	y	y'	δ	Scatter Type

Table D.10: Column headings for the `TrackingOutputAV` output file.

D.4 Physics Processes

The `ParticleBunchProcess` is a template `BunchProcess`. This can represent any mechanism, for example particle tracking, application of a physical process (*e.g. synchrotron radiation*), or even data output. The process may be run at selected `Accelerator Component`s in the lattice, or all of them, and may be defined for only a single type of bunch, or all. These processes (henceforth referred to as physics processes) may specify a priority (where 0 is highest, followed by incremental integers), and by default will be executed in the priority that they are added to the tracker. Particle tracking is a `Bunch Process` and by default takes priority over all other physics processes.

Each physics process must define a number of functions; `InitialiseProcess()` initialises the process with the particle bunch, `SetCurrentComponent()` allows the user to set the current accelerator component and verify if the process should be run for this component, `GetMaxAllowedStepSize()` returns the step size for the process, and the main process is normally contained within `DoProcess()`. Processes can have more functions as required, but these are the minimum.

Figure D.25 shows an example of the use of the `CollimateProtonProcess` in MERLIN. The contents of this are detailed in Chapter 4.

A complete example of the method of creating the HEL process is shown in Fig. D.26, the contents of which are described in Chapter 4.

D.5 Collimation Outputs

A number of output functions have been constructed to obtain useful data from collimation simulations. The majority of these are part of the `ScatteringModel` class, and are detailed in this section.

D.5.1 `Death Report`

`DeathReport()` is a function of the `ScatteringModel`, meaning that it can only record data from within the collimator. It is used to store special forms of output, as a template function for the user. For example in its simplest use it can store data for

```
CollimateProtonProcess* myCollimateProcess = new CollimateProtonProcess
    (2, 4, NULL);

// Create LossMapDustbin and attach to process
LossMapDustbin* myLossMapDustbin = new LossMapDustbin;
  myCollimateProcess->SetDustbin(myLossMapDustbin);
  // Use Beam2() function only when running LHC beam 2
  bool beam2 = true;
  myLossMapDustbin->Beam2(beam2);

// Create a ScatteringModel, associated outputs, and attach to process
ScatteringModel* myScatter = new ScatteringModel;
  // Use default MERLIN (true) or ST-like (false) composite materials
      method
  myScatter->SetComposites(true);
  // Output proton impacts on the primary collimator
  myScatter->SetJawImpact("TCP.C6R7.B2");
  // Output shower plot data for primary collimator scattering
  myScatter->SetScatterPlot("TCP.C6R7.B2");
  // Output exact map of losses for secondary collimator
  myScatter->SetJawInelastic("TCSG.B5R7.B2");
  // Attach ScatteringModel to process
  myCollimateProcess->SetScatteringModel(myScatter);

// Create FlukaLosses and attach to process
FlukaLosses* myFlukaLosses = new FlukaLosses;
  myCollimateProcess->SetFlukaLosses(myFlukaLosses);

// Set predefined combination of ScatteringProcesses
bool use_sixtrack_like_scattering = false;
if(use_sixtrack_like_scattering){myScatter->SetScatterType(0);}
else{myScatter->SetScatterType(4);}

// Specify 10 cm bins for collimation aperture check and non-collimator
    loss output
myCollimateProcess->SetOutputBinSize(0.1);

// Attach the collimation process to the tracker
myParticleTracker->AddProcess(myCollimateProcess);
```

Figure D.25: A complete example of the `CollimateProtonProcess` and associated classes and functions that are used in MERLIN to define the collimation simulation.

collimator-only loss maps.

```
// Constructor defining: priority, mode, HEL current, electron beta,
// proton beam rigidity, and effective HEL length
HollowELensProcess* myHELProcess = new HollowELensProcess(3, 1, 5,
    0.195, 2.334948339E4, 3.0);

  // Set HEL beam direction, 1 = opposite to protons (focussing)
  myHELProcess->SetElectronDirection(1);

  // Select radial (measured) or perfect profile
  // using previously defined boolean
  if(perfect_profile){  myHELProcess->SetPerfectProfile();    }
  else{                 myHELProcess->SetRadialProfile();
  // Scale the measured radial profile empirically for LHC hardware
                        myHELProcess->SetLHCRadialProfile();  }

  // Centre the HEL on the closed orbit and use the lattice functions
  // to match the HEL beam size to the proton beam envelope
  // using minimum radius = 4 sigma and maximum radius = 8 sigma
  myHELProcess->SetRadiiSigma(4, 8, myAccModel, emittance, emittance,
      myTwiss, 7000);

  // Select current modulation mode using previously defined booleans
  if(ACon){             myHELProcess->SetAC(0.31, .002, 5E-5, 1E3, 2.);
                        myHELProcess->SetOpMode(AC);        }
  else if(DCon){        myHELProcess->SetOpMode(DC);        }
  else if(Diffusiveon){ myHELProcess->SetOpMode(Diffusive); }
  else if(Turnskipon){  myHELProcess->SetTurnskip(31);
                        myHELProcess->SetOpMode(Turnskip);  }

  // Select geometrical operation mode using previously defined booleans
  if(elliptical_HEL){     myHELProcess->SetEllipticalMatching(true);  }
  else if(hula_HEL){      myHELProcess->SetHulaElliptical(true);      }
  else if(closehula_HEL){ myHELProcess->SetCloseHulaElliptical(true); }
  else if(pogo_HEL){      myHELProcess->SetPogoElliptical(true);      }

// Attach the process to the tracker
myParticleTracker->AddProcess(myHELProcess);
```

Figure D.26: A complete example of the `HollowELensProcess` and associated setting functions that are used in MERLIN to define the hollow electron lens simulation.

D.5.2 Impact Parameter

The `ScatteringModel::JawImpact()` function outputs the co-ordinates of particles that impact the front face of selected collimators. As well as co-ordinates, the turn at which the impact occurred is also output, this allows the user to observe any change in the impact parameter over time. This output is useful for observing the effect of the

HEL. An example of this output for a single turn in a collimation simulation is shown in Fig. 4.11, showing the correspondence between the initial distribution (which starts immediately in front of the primary collimator), with the recorded impact co-ordinates.

ID	x	x'	y	y'	ct	δ	**Turn**

Table D.11: Column headings for the `JawImpact` output file.

D.5.3 Scattering Tracker

The `ScatteringModel::ScatterPlot()` function stores the position of particles at each path length step along a collimator to plot scattering tracks along the collimator. This function has been a useful tool for debugging the collimation process, ensuring that aperture checks are performed at appropriate intervals, and showing the effect of the collimation bin size. The columns of this output file are shown in Table D.12.

ID	z	x	y	**Turn**

Table D.12: Column headings for the `ScatterPlot` output file.

D.5.4 Inelastic Interactions

`ScatteringModel::JawInelastic()` stores the co-ordinates of inelastic interactions, in the format shown in Table D.13.

ID	x	x'	y	y'	ct	δ	z	**Turn**

Table D.13: Column headings for the `JawInelastic` output file.

D.5.5 Selected Interactions

`ScatteringModel::SelecScatter()` stores the co-ordinates of selected interactions, outputting the data shown in Table D.14, which includes the momentum transfer t and the polar angle θ.

As the raw data files produced can be very large, the `OutputSelectScatterHistogram()` function was created to histogram the data and produce a smaller output. An example of the histogrammed polar angle data is shown in Fig. D.27.

ID	x	x'	y	y'	ct	δ	z	θ	t	type

Table D.14: Column headings for the `SelectScatter` output files.

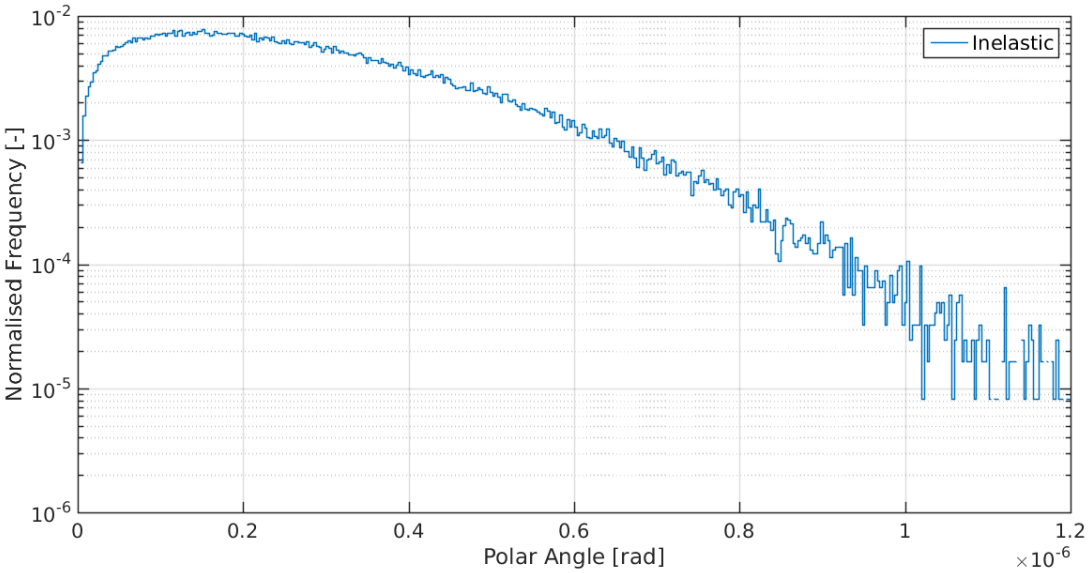

Figure D.27: Polar angle histogrammed in the `OutputSelectScatterHistogram()` output function, showing the angular distribution of particles that have undergone inelastic interactions in a collimator made of CFC AC150K®. In reality this angular spread is given by MCS.

D.5.6 `Dustbin`

`Dustbin` is a virtual class used to store lost particle data. It uses the `LossData` struct shown in Table D.15 to store information that may be useful for output.

The `Dispose()` function is called when a particle is lost, and a `LossData` struct is created and stored in a vector. The `Finalise()` function is called at the end of the simulation, and is used to collate losses, which are stored in a separate vector. The user may define multiple methods for this, the constructor takes an `OutputType` to switch between these. To output the collated data, the `Output()` function takes a user defined output stream (pointer) and performs the previously selected output.

D.5.6.1 `LossMapDustbin`

The `LossMapDustbin` is a child class of the `Dustbin` output class which was created for loss map production, the current available options for loss maps are to store losses

Name	Variable Type	
ElementName	string	Name of element
p	PSvector	Lost particle
s	double	s position of element
interval	double	Bin interval within element
position	double	Exact loss position
length	double	Length of element
lost	double	Number of lost particles
temperature	int	Warm/cold/collimator element
turn	int	Turn in which particle is lost
coll_id	int	Collimator ID
angle	double	Collimator tilt angle

Table D.15: Information stored in the `LossData` struct.

to the nearest element, at the precise position of the loss (in the s co-ordinate), or in 10 cm bins (the default setting).

The `LossMapDustbin::Dispose()` function takes the current `AcceleratorComponent` address, the exact position of the loss, the particle address, and the turn. It is called in the `CollimateProtonProcess::DoScatter()` function when a particle is lost in a collimator, and the `CollimateParticleProcess::DoCollimation()` function when a particle is lost in a non-collimator element. Only collimator losses may be output at exact positions, as the method used for calculating the position of non-collimator losses uses the user specified bin size (which by default allows the location of the non-collimator loss to the nearest 10 cm). This is also the case when scattering is turned off for the collimation process (collimators are treated as black absorbers), as the exact loss position for a collimator is the s co-ordinate of an inelastic interaction or energy cut, whereas when no scattering takes place, the location is where a particle exits the aperture.

The temperature indicator is used to output cold, warm, and collimator losses in different colours or separately. The `LossMapDustbin` contains a hard-coded selection method to decide whether the element is superconducting (cold), normal conducting (warm), or a collimator, that is only relevant for the LHC. The user may use the `LossMapDustbin::Beam2()` function to switch this selection function when performing simulations for beam 2, this is necessary as the selection uses element names and positions. Though this hard-coded method is not ideal, the `Dustbin` class is designed to provide the user the means by which to define, store, collate, and print their own output with full control.

`LossMapDustbin::Finalise()` is optimised by using the `LossData` struct, which has overloaded functions for the relational $<$ and $=$, binary $+$, and unary $++$ operators, used to sort, collate, and increment losses. The column headings of the standard 10 cm loss map output are specified in Table D.16.

Element Name	s+**Interval**	**Interval**	n_{lost}	**Temperature**	**Collimator Length**

Table D.16: Column headings for the `LossMapDustbin` output file.

D.5.7 `FlukaLosses`

`FlukaLosses` was created to record every interaction of a tracked proton with all collimator jaws. The class imitates the Fluka output of SixTrack [111], storing all inelastic and single diffractive interactions. As with the `FLUKADatabase` this class is not currently used with FLUKA but provides a method of direct comparison with current SixTrack outputs. The class also allows selection of other interactions to record, however for all interactions other than inelastic, the same particle may interact more than once.

In a similar manner to the `Dustbin` class, a `FlukaLossData` struct, detailed in Table D.18, has overloaded operator functions and is used to store interaction data.

Variable Type	**Name**	
int	coll_id	Collimator ID
double	angle	Collimator tilt angle
double	s	s position of element
PSvector	p	Lost particle
int	turn	Turn in which particle is lost
double	lost	Number of lost particles
double	x_offset	Collimation plane offset (orbit)
double	y_offset	Perpendicular plane offset (orbit)
string	ElementName	Name of element

Table D.17: Information stored in the `FlukaLossData` struct.

`FlukaLosses::Record()` is similar to `Dustbin::Dispose()`, and creates a `FlukaLossData` struct that is stored in a vector. This class also has a `Finalise()` function that may be used to collate the stored data, and store it in another vector.

The `FlukaLosses::Output()` function strays from the unit conventions of MERLIN, outputting the particle displacements x, y in mm, and angles x', y' in mrad. The

Name	Variable Type	
coll_id	int	Collimator ID
angle	double	Collimator tilt angle
s	double	s position of element
p	PSvector	Lost particle
turn	int	Turn in which particle is lost
lost	double	Number of lost particles
x_offset	double	Collimation plane offset (orbit)
y_offset	double	Perpendicular plane offset (orbit)
ElementName	string	Name of element

Table D.18: Information stored in the `FlukaLossData` struct.

column headings for the output file are stated in Table D.19.

Collimator ID	α	s	$x - x_{offset}$	x'	$y - y_{offset}$	y'	**Type**	**Particle ID**	**Turn**

Table D.19: Column headings for the `FlukaLosses` output file.

References

[1] The HL-LHC Collaboration. **FP7 High Luminosity Large Hadron Collider Design Study**. Technical report, CERN, 2010. 4

[2] R. Bruce et al. **LHC Collimation**. In *4th Annual NGACDTConference*, 2015. 4

[3] Guillame Robert-Demolaize. *Design and performance optimisation of the LHC collimation system*. PhD thesis, Ecole Nationale Superiore de Physique de Grenoble, November 2006. 4, 17, 82, 93, 152, 231

[4] Gianluca Valentino. *Fast Automatic Beam-Based Alignement of the LHC Collimator Jaws*. PhD thesis, University of Malta, June 2013. 15, 48, 51, 106

[5] Fermilab. **The Tevatron**. `http://www.fnal.gov/pub/tevatron/`. 50, 61

[6] CERN. **LHC Accelerator and Performance Statistics**. `https://acc-stats.web.cern.ch/acc-stats/#lhc/`. 50

[7] L. Rossi. **Status of HiLumi: from Design Study to Construction Project**. In *HiLumi-LARP*, 2015. 15, 50

[8] A. Lechner. **Energy deposition with cryo-collimators in IR2 (ions) and IR7**. In *HiLumi-LARP*, 2013. 16, 51, 61, 62

[9] O. Brüning and L. Rossi. *The High Luminosity Large Hadron Collider: The New Machine for Illuminating the Mysteries of the Universe*. World Scientific, 2015. 15, 16, 40, 52, 53, 54, 55, 56, 58, 59, 60, 62, 81

[10] F. Bordry. **CERN Plan for Accelerator & Schedule**. In *HiLumi-LARP*, 2015. 53

[11] S. Fartoukh. **First demonstration with beam of the Achromatic Telescope Squeezing (ATS) scheme**. Technical report, CERN-ATS-2012-069, 2012. 55, 56

[12] S. Fartoukh et al. **The 10 cm beta* ATS MD**. Technical report, CERN-ATS-Note-2013-004 MD, 2013. 56

[13] V. Shiltsev et al. **Summary of the Working Group on Single Particle Effects: Parasitic Long-Range Beam-Beam Interactions**. Technical report, CERN-2014-004, pp.253-255, 2014. 15, 57

[14] KEK. **KEKB Accelerator**. `http://www-acc.kek.jp/KEKB/`. accessed 2016. 58

[15] SLAC. **Stanford Linear Accelerator Center**. `https://www6.slac.stanford.edu/`. accessed 2016. 59

[16] F. Carra et al. **LHC collimators with embedded beam position monitors: a new advanced mechanical design**. In *IPAC*, 2011. 59

[17] S. Bruce et al. **HL-LHC Deliverable Report D5.3: Beam Halo Simulations**. Technical report, CERN-ACC-2013-008, 2012. 15, 61

[18] CERN BE-ABP. **SixTrack: Single Particle Tracking Code**. `http://frs.home.cern.ch/frs/`. accessed 2016. 61, 109, 120, 123

[19] FLUKA Team. **FLUKA**. `http://www.fluka.org/fluka.php`. accessed 2016. 61, 120, 175, 219

[20] G. Stancari et al. **Collimation with Hollow Electron Beams**. *Phys. Rev. Lett.*, **107**(084802), 2011. 61, 108

[21] J. Wagner et al. **Active halo control through narrow-band excitation with the ADT**. Technical report, CERN-ACC-NOTE-2016-0009, 2016. 63

[22] Daniele Mirarchi. *Crystal Collimation for LHC*. PhD thesis, Imperial College London, March 2015. 63, 231

[23] R. Rossi et al. **Crystal Collimation MD 333**. LHC Studies Working Group. 63

[24] A. Wolski. *Beam Dynamics in High Energy Particle Accelerators*. Imperial College Press, 2014. 67, 68, 71, 73, 79

[25] R. Appleby. **Beam Dynamics**. Cockcroft Lectures 2011. 68, 69

[26] H. Wiedemann. *Particle Accelerator Physics I: Basic Principles and Linear Beam Dynamics*. Springer, 1999. 68

[27] S. Y. Lee. *Accelerator Physics*. World Scientific, 2004. 69, 72

[28] D. C. Karey et al. **Third-Order TRANSPORT with MAD Input: A Computer Program for Designing Charged Particle Beam Transport Systems**. Technical report, Fermilab-Pub-98-310, 1998. 78

[29] R. Bruce et al. **Sources of machine-induced background in the ATLAS and CMS detectors at the CERN Large Hadron Collider**. *Nucl. Instrum. Meth. A*, **729**(825-840), 2013. 81

[30] G. Valentino et al. **Beam diffusion measurements using collimator scans at the LHC**. *Phys. Rev. ST Accel. Beams*, **16**(021003), 2013. 82

[31] H. Burkhardt et al. **Collimation down to 2 sigmas in special physics runs in the LHC**. In *IPAC*, 2013. 82

[32] Chiara Bracco. *Commissioning scenarios and tests for the LHC collimation system*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2009. 17, 40, 41, 43, 82, 88, 89, 96, 98, 99, 100, 101, 322, 323, 324

[33] R. Bruce et al. **Collimation status and layout for IR1 and IR5**. In *HiLumi-LARP*, 2014. 18, 82, 103, 105, 107

[34] B. Dehning. **Beam Loss Monitors at LHC**. Technical report, CERN Yellow Report CERN-2016-002, 2016. 83

[35] B. Dehning et al. **Overview of LHC Beam Loss Measurements**. Technical report, CERN-ATS-2011-064, 2011. 84

[36] J. Wenninger. **Machine Protection and Operation for LHC**. In *Joint Interational Accelerator School on Machine Loss and Beam Protection, Newport Beach*, 2014. 84, 85, 88

[37] B. Dehning et al. **The LHC beam loss measurement system**. In *PAC*, 2007. 86

[38] J. Kerby et al. **INNER TRIPLET QUADRUPOLE MQXB**. Technical report, CERN Functional Specification: LHC-LQX-ES-0002, 2001. 86

[39] O. Bruning et al. **Low Gradient Triplet Magnets**. In *LHC Lumi*, 2006. 86

[40] M. Lamont. **LHC Performance in Run 2 and Beyond**. 27th International Symposium on Lepton Photon Interactions at High Energies, 17th Aug 2015. 17, 86

[41] Florian Burkart. *Beam Loss and Beam Shape at the LHC Collimators*. PhD thesis, Goethe Universiat Frankfurt, March 2012. 87

[42] D. Wollmann et al. **Status of Machine Protection Studies**. In *HiLumi-LARP*, 2014. 87

[43] E. Quaranta et al. **Updates of Irradiation Tests at GSI on LHC Collimator Materials**. 42nd LHC Collimation Upgrade Specification Meeting, 1st Aug 2014. 90

[44] E. Bertarelli et al. **Novel Materials for Collimators at LHC and its Upgrades**. In *ICFA Advanced Beam Dynamics Workshop on High-Intensity and High-Brightness Hadron Beams*, 2014. 40, 90, 91

[45] E. Quaranta et al. **Collimation Cleaning at the LHC with Advanced Secondary Collimator Materials**. In *IPAC*, 2015. 41, 44, 90, 131, 132, 133, 175, 331, 332, 334, 337

[46] A. Ryazanov. **Investigations of fast proton irradiation effects on Mo-Diamond collimator materials for LHC at NRC Kurchatov Institute and comparison with the results for Co-Diamond materials**. 43rd LHC Collimation Upgrade Specification Meeting, 8th Aug 2014. 17, 91

[47] Nicola Mariani. *Development of Novel, Advanced Molybdenum-based Composites for High Energy Physics Applications*. PhD thesis, Politecnico di Milano, 2014. 40, 90, 91, 284, 332, 334

[48] S. Redaelli et al. **Collimation Upgrade Path for HL-LHC**. In *HiLimi-LARP*, 2014. 90, 91, 103

[49] James Molson. *Proton scattering and collimation for the LHC and LHC luminosity upgrade*. PhD thesis, University of Manchester, 2014. 24, 38, 40, 91, 123, 124, 135, 145, 146, 147, 149, 152, 155, 156, 158, 185, 187, 188, 190, 198, 199, 334, 346, 348

[50] J. B. Jeanneret. **Optics of a two-stage collimation system**. *Phys. Rev. ST Accel. Beams*, **1**(081001), 1998. 94

[51] Mike Seidel. *The proton collimation system of HERA*. PhD thesis, Universitat Hamburg, June 1994. 17, 94

[52] S. Redaelli et al. **Operational performance of the LHC collimation system**. In *HB2010*, 2010. 95

[53] S. Redaelli et al. **Analysis of failures of the LHC collimators during the 2010-2013 operation**. In *IPAC*, 2013. 41, 95, 101, 102, 147

[54] LHC Collimation Group. **LHC Collimation Activities during LS1 and commissioning**. `http://lhc-collimation-project.web.cern.ch/lhc-collimation-project/LS1/default.php`. 95

[55] S. Redaelli et al. **Overview of LS1 collimation project activities**. 170th Meeting of the LHC Collimation Working Group, 3rd Feb 2014. 40, 97, 98

[56] The ATLAS collaboration. **Characterisation and mitigation of beam-induced backgrounds observed in the ATLAS detector during the 2011 proton-proton run**. *IOP Journal of Instrumentation*, **8**(P07004), 2013. 17, 101

[57] S. Redaelli et al. **LHC Collimation for Run II and beyond**. 57th ICFA Advanced Beam Dynamics Workshop on High-Intensity and High-Brightness Hadron Beams, 3rd July 2016. 102

[58] H. G. Morales et al. **Collimation IR Layout for the Incoming Beam**. In *HiLimi-LARP*, 2014. 103

[59] E. Hoyer et al. **LHC IP1 IP5 Neutral Beam Absorbers (TAN)**. Technical report, CERN Functional Specification: LHC-TAN-ES-0200, 1999. 103

[60] F. Cerutti et al. **Energy Deposition for HL-LHC v1.1: TAN/D2/Q4**. In *HiLimi-LARP*, 2014. 103, 104

[61] J. Jowett. **Dispersion Suppressor Collimators for Heavy-Ion Operation**. In *HiLimi-LARP*, 2014. 104

[62] F. Carra et al. **Status of TCLD Design and Integration**. In *HiLimi-LARP*, 2014. 17, 104

[63] D. Perini et al. **Advanced Collimation Design Studies (TCT with wire, hollow e lens)**. In *HiLumi-LARP*, 2014. 18, 105, 106

[64] V. Shiltsev et al. **Experimental demonstration of compensation of beam-beam effects by electron lenses**. *Phys. Rev. Lett.*, **99**(244801), 2007. 108

[65] V. Shiltsev. **On Possible Use of Electron Lenses in LHC**. Technical report, FERMILAB-CONF-06-505-AD, 2006. 108, 109

[66] Zhang et al. **The Origination and Diagnostics of Uncaptured Beam in the Tevatron and Its Control by Electron Lenses**. *Phys. Rev. ST Accel. Beams*, **11**(051002), 2008. 108

[67] V. D. Shiltsev. *Electron Lenses for Particle Colliders*. Springer, 2015. 108, 110, 111, 113

[68] V. Previtali et al. **Numerical simulations of a proposed hollow electron beam collimator for the LHC upgrade at CERN**. Technical report, FERMILAB-TM-2560-APC, 2013. 30, 109, 124, 164, 231, 236, 240, 241

[69] The HL-LHC Collaboration. **HL-LHC Preliminary Design Report**. Technical report, CERN-ACC-2014-0300, 2014. 109

[70] G. Stancari et al. **Conceptual design of hollow electron lenses for beam halo control in the Large Hadron Collider**. Technical report, FERMILAB-TM-2572-APC, 2014. 109, 246

[71] R. Bruce et al. **Simulations and measurements of beam loss patterns at the CERN Large Hadron Collider**. *Phys. Rev. ST Accel. Beams*, **17**(081004), 2014. 19, 38, 120, 121, 219, 225, 325

[72] ANSYS Inc. **ANSYS-Autodyn**. `http://www.ansys.com/Products/Structures/ANSYS-Autodyn`. accessed 2016. 120

[73] Linear Collider Collaboration. **International Linear Collider**. `https://www.linearcollider.org/ILC`. 123

[74] F. Poirier et al. **An ILC Main Linac Simulation Package Based on MERLIN**. In *EPAC*, 2006. 123

[75] D. Kruecker et al. **MERLIN-Based Start-To-End Simulations of Luminosity Stability for the LHC**. In *PAC*, 2007. 123

[76] D. Kruecker et al. **Simulation Studies on Coupler Wakefield and RF Kicks for the International Linear Collider with MERLIN**. In *EPAC*, 2008. 123

[77] R. J. Barlow et al. **The MERLIN Simulation Program: New Features Used in Studies of the LHC Collimation System Using MERLIN**. In *IPAC*, 2011. 123

[78] A. Toader et al. **Simulation of the LHC Collimation System Using MERLIN**. In *PAC*, 2009. 123

[79] J. G. Molson et al. **Simulating the LHC Collimation System with the Accelerator Physics Library MERLIN, and Loss Map Results**. In *IPAC*, 2012. 123, 124

[80] M. Serluca et al. **Comparison of MERLIN/SixTrack for LHC Collimation Studies**. In *IPAC*, 2014. 123

[81] T. Trenkler and J. B. Jeanneret. **K2, a software package evaluating collimation systems in circular colliders (manual)**. Technical report, CERNSL/94105 (AP), 1994. 123, 175

[82] G. Robert-Demolaize et al. **A New Version of SixTrack with Collimation and Aperture Interface**. In *PAC*, 2005. 123

[83] H. Rafique et al. **Simulation of Hollow Electron Lenses as LHC Beam Halo Reducers using MERLIN**. In *IPAC*, 2015. 124

[84] H. Rafique et al. **MERLIN for LHC Collimation**. In *ICAP*, 2015. 124

[85] CERN BE-ABP. **MADX: Methodical Accelerator Design**. `http://cern.ch/madx`. accessed 2012. 126, 342

[86] James Molson. **Private Communication**. 127, 158

[87] Particle Data Group. **Passage of particles through matter**. `http://durpdg.dur.ac.uk/lbl/index.htmll`. 2005. 147, 155, 156, 157, 335

[88] Nuria Catalan Lasheras. *Transverse and longitunidal beam collimation in a high-energy proton collider (LHC)*. PhD thesis, Universidad de Zaragoza, November 1998. 147, 153, 155

[89] K. Goulianos. **Diffractive interactions of hadrons at high energies**. *Physics Reports*, **101(3):169-219**, 1983. 148

[90] J. Berinder et al. (Particle Data Group). **Diffractive interactions of hadrons at high energies**. *Eur. Phys. J. C*, **86**(010001), 2012. 20, 156

[91] Claudia Tambasco. *An Improved Scattering Routine for Collimation Tracking Studies at LHC*. Master's thesis, University of Rome Sapienza, September 2014. 158, 197, 198

[92] Vince Moens. *Experimental and numerical studies on the proposed application of hollow electron beam collimation for the LHC at CERN*. Master's thesis, Ecole Polytechnique Federale de Lausanne, September 2013. 163

[93] Elena Quaranta. **Private Communication**. 32, 33, 41, 176, 187, 259, 260, 261, 262, 332

[94] R. Aßman et al. **Tools for Predicting Cleaning Efficiency in the LHC**. In *PAC*, 2003. 22, 175, 177, 178, 199

[95] Alessandra Valloni. **Private Communication**. 27, 187, 209, 214, 215, 216, 217, 218, 224

[96] R. B. Appleby et al. **The Practical Pomeron for High Energy Proton Collimation**. *Phys. Rev. D*, 2016. 199

[97] A. Valloni et al. **MERLIN Cleaning Studies with Advanced Collimator Materials for HL-LHC**. In *IPAC*, 2016. 199

[98] S. Tygier. **Merlin 4 TeV squeeze simulation status**. 74th LHC Collimation Upgrade Specification Meeting, 24th June 2016. 220

[99] H. G. Morales et al. **HL-LHC off-momentum cleaning simulations**. 63rd LHC Collimation Upgrade Specification Meeting, 18th Sep 2015. 279

[100] G. Stancari et al. **Measurements of beam halo diffusion and population density in the Tevatron and in the Large Hadron Collider**. 54th

ICFA Advanced Beam Dynamics Workshop on High-Intensity, High-Brightness, and High-Power Hadron Beams, November 10th 2014. 315

[101] G. Valentino et al. **Halo population and diffusion measurements with 300 bunches**. 74th LHC Collimation Upgrade Specification Meeting, 24th Jun 2016. 315

[102] CERN. **High Radiation to Materials Facility**. `https://espace.cern.ch/hiradmat-sps/Wiki%20Pages/Home.aspx`. 318

[103] E. Quaranta et al. **Beam Losses on TCLD During Fast Failures as Input for Material Choices**. 68th LHC Collimation Upgrade Specification Meeting, 26th Feb 2016. 318

[104] David Brett. *Precision element modelling for long term tracking in the LHC luminosity upgrade*. PhD thesis, University of Manchester, 2014. 320

[105] R. Assmann et al. **Requirements for the LHC collimation system**. Technical report, CERN-LHC-PROJECT-REPORT599, 2002. 43, 323

[106] CERN Editorial Board. **LHC Design Report, Volume I: The LHC Main Ring**. Technical report, CERN-2004-003, 2004. 43, 324

[107] R. Assmann et al. **LHC collimation: design and results from prototyping and beam tests**. In *PAC*, 2005. 325

[108] S. M. Seltzer et al. **Evaluation of the collision stopping power of elements and compounds for electrons and positrons**. *The International Journal of Applied Radiation and Isotopes*, **33(11):1189-1218**, 1982. 334

[109] CERN. **Future Circular Collider**. `https://fcc.web.cern.ch/Pages/default.aspx`. 352

[110] J. Molson et al. **Simulation of the FCC-hh Collimation System**. In *IPAC*, 2016. 352

[111] A. Mereghetti et al. **SixTrack-Fluka Active Coupling for the Upgrade of the SPS Scrapers**. In *IPAC*, 2013. 368